



Solução de Business Intelligence para Seguros

António Roberto Taveira de Vasconcelos Pinto
de Gouveia

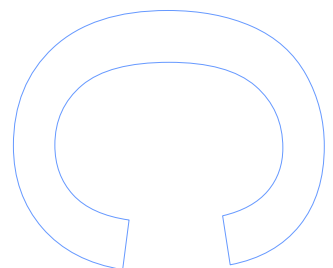
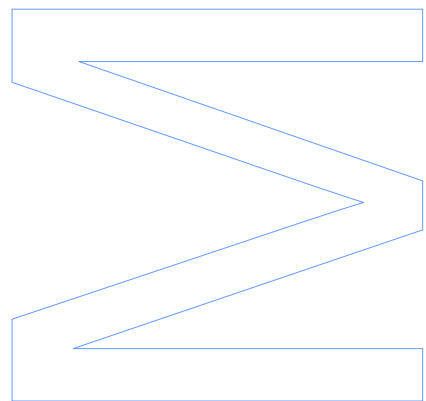
Mestrado Integrado em Engenharia de Redes e Sistemas Informáticos
Departamento de Ciências de Computadores
2013

Orientadores

Eng. Margarida Mesquita e Eng. André Ferreira

Coorientador

Professora Doutora Ana Paula Tomás

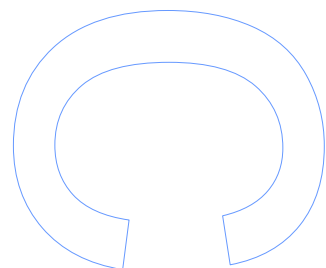
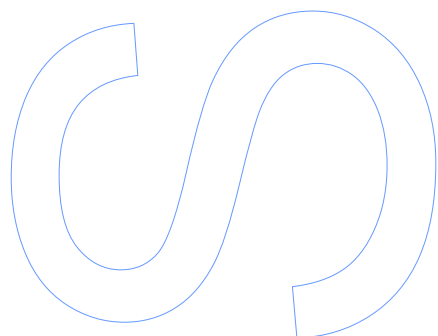
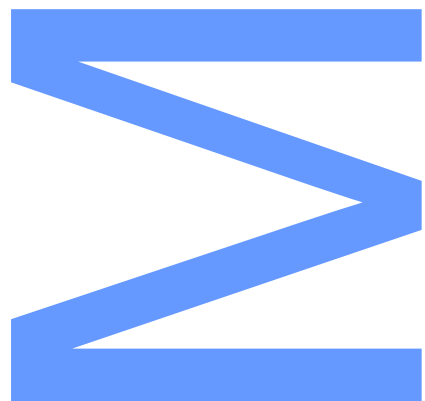




Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ____/____/____



Resumo

Nos dias de hoje as empresas têm necessidade de tratar todo um manancial de informação, a que têm acesso, pelas mais variadas vias, para as suas tomadas de decisão. A recolha desta informação qualitativa e crucial, se bem tratada e integrada, permite-lhes acumular "inteligência" para competir em mercados cada vez mais exigentes. Esta inteligência de negócios, traduzida do inglês *Business Intelligence* (BI), podemos explicá-la como um método que visa ajudar as empresas a tomar as decisões inteligentes, através do acesso a dados e informação recolhida dos diversos sistemas de informação.

Com este relatório pretende-se mostrar a implementação de uma estrutura de BI, aplicada à atividade seguradora.

Apresentamos um projeto desenvolvido na empresa i2S no contexto dum estágio, que consistiu no desenvolvimento de um Data Warehouse e de um conjunto de relatórios e métricas de gestão aplicadas na indústria seguradora.

Conteúdo

Resumo	1
1 Introdução	5
1.1 Objetivo do Projeto de Estágio	6
1.2 Estrutura do Relatório	6
2 Business Intelligence e Data Warehousing	7
2.1 Business Intelligence	7
2.1.1 Sistema de Business Intelligence	7
2.1.2 Capacidades de Business Intelligence	8
2.1.3 Benefícios de Business Intelligence	11
2.2 Data Warehouse	11
2.2.1 Data Mart	13
2.2.2 Processo de Data Warehousing	15
2.2.3 Processo ETL	15
2.2.4 Modelo Dimensional	16
3 Ferramentas Utilizadas no Projeto	21
3.1 Plataformas de Business Intelligence	21
3.1.1 MicroStrategy	21
3.1.2 Porquê MicroStrategy?	21
3.1.3 Componentes da Plataforma MicroStrategy (MSTR)	25

3.1.4	In-memory ROLAP	26
3.2	Ferramentas de ETL	27
3.2.1	Comparação entre Ferramentas de ETL <i>Open Source</i>	27
4	Análise do Projeto	29
4.1	Análise do Departamento BI da i2S	29
4.2	Análise dos Sistemas Operacionais e do Informacional	31
4.3	Arquitetura do Data Warehouse	32
4.3.1	Arquitetura de Data Marts Independentes	33
4.3.2	Arquitetura Data Warehouse em <i>Bus</i> (DWB)	33
4.3.3	Arquitetura <i>Hub and Spoke</i>	34
4.3.4	Arquitetura Federada	35
4.3.5	Abordagem Híbrida	36
4.4	Arquitetura Proposta para Este Projeto	36
4.5	Levantamento de Requisitos	37
5	Modelação Dimensional	39
5.1	Processos de Negócio	39
5.2	Definição da Granularidade	40
5.3	Identificação das Dimensões	42
5.4	Matriz do Data Warehouse em <i>Bus</i> e Dimensões Conformes	46
5.5	Identificação dos Factos	49
6	Desenvolvimento do Data Warehouse	53
6.1	Tabelas de Factos Transacionais	53
6.2	Tabelas de Factos de Sumarização Periódica	56
6.3	Tabelas Agregadas	57
6.4	Tabelas de Dimensão	58
6.5	Tabela de Objetivos e KPIs	62

6.6	Esquema em Floco de Neve	64
7	Desenvolvimento do Processo de Carregamento	66
7.1	Carregamento Total da Tabela de Dimensão de Região	69
7.2	Carregamento da Tabela de Dimensão de Titular	70
7.3	Carregamento da Tabela de Factos das Apólices	71
7.4	Carregamento Total Mensal da Tabela de Dimensão de Estrutura Comercial . .	72
8	Demonstração na Plataforma MicroStrategy	75
9	Conclusão e Trabalho Futuro	82
9.1	Conclusão	82
9.2	Trabalho Futuro	83
A	Acrónimos	86
B	Tabelas de Factos de Inventário	87
C	Métricas e Relatórios Desenvolvidos no MSTR	89
	Referências	94

Capítulo 1

Introdução

Atualmente assiste-se a um crescendo de quantidade de informação disponibilizada pelos sistemas de informação das empresas. Tão elevada quantidade de informação só ajudará a otimizar as decisões, se e só se for apresentada de uma forma organizada e adequada (*user-friendly*), com um bem estruturado e rápido sistema de busca, em sistemas de gestão de informação, hoje conhecidos por solução de *Business Intelligence* (BI).

A atividade seguradora em Portugal e na exploração dos principais ramos não vida, nomeadamente Acidentes de Trabalho e Automóvel, tem apresentado, nos últimos anos, uma assinalável degradação, apresentando uma expressiva redução de produção (volume de prémios) e resultados técnicos muito deficitários.

A progressiva diminuição de prémios, resulta da atual crise económica e da consequente contração dos capitais a segurar e da redução dos preços praticados, dada a forte concorrência do mercado.

Neste contexto tão agressivo e exigente as empresas de seguros terão de dotar-se de um sistema de gestão e controlo da sua informação (soluções de BI), com históricos completos e consistentes, pois só assim poderão definir políticas de subscrição e de tarifação (preços) rigorosas que lhes permitirão restabelecer o equilíbrio dos resultados.

O relatório que se apresenta foi elaborado durante um estágio na i2S (Insurance Software Solutions).

A i2S é uma *software house*, com 28 anos de atividade no mercado global de seguros, desenvolvendo soluções informáticas em várias partes do mundo, para os diferentes intervenientes da atividade seguradora: Companhias de Seguros Vida, Não Vida, Mediadores/Corretores, Peritos e Sociedades Gestoras de Fundos de Pensões.

A i2S pretende desenvolver uma solução de BI para seguros sobre os sistemas de bases de dados que tem sob gestão.

1.1 Objetivo do Projeto de Estágio

Como objetivo deste estágio, teve de se desenvolver uma solução de BI para seguros sobre os sistemas de bases de dados usados nesta empresa, utilizando para isso todos os conceitos já identificados bem como um processo de ETL já existente. No desenvolvimento desta solução utilizou-se uma plataforma de BI que tem muitas das capacidades de manipulação e exploração inerentes a uma ferramenta desta natureza, construindo-se alguns relatórios e métricas base de gestão, na área de seguros não vida. Com este sistema avançado de análise de informação, o utilizador final terá autonomia total para construir os seus gráficos, *reports* e análises de gestão personalizadas.

1.2 Estrutura do Relatório

Este relatório está estruturado da seguinte forma:

- **Capítulo 1:** breve introdução que inclui a descrição do tema do projeto de estágio e da estrutura do relatório.
- **Capítulo 2:** apresentação de alguns conceitos de *Business Intelligence* e de *Data Warehousing*.
- **Capítulo 3:** apresentação das ferramentas utilizadas neste projeto.
- **Capítulo 4:** análise do departamento de *Business Intelligence* da i2S, dos sistemas de origem, da arquitetura do DW definida para este projeto e descrição do levantamento de requisitos efetuado.
- **Capítulo 5:** descrição do processo de conceção de vários data marts.
- **Capítulo 6:** descrição do desenvolvimento do data warehouse.
- **Capítulo 7:** descrição do desenvolvimento do processo de carregamento de dados no data warehouse.
- **Capítulo 8:** apresentação de uma pequena demonstração na plataforma MicroStrategy.
- **Capítulo 9:** conclusões finais.

Capítulo 2

Business Intelligence e Data Warehousing

2.1 Business Intelligence

Se as empresas quiserem sobreviver e crescer, neste mundo tão competitivo, terão de ter um acesso rápido e facilitado à informação sobre o seu negócio para conseguirem entendê-lo. Só assim as empresas poderão (re)agir em tempo oportuno, inovar e descobrir oportunidades. Com esta necessidade premente, as empresas poderão recorrer a Business Intelligence que pode ser definido como um "termo genérico para designar aplicações, infraestruturas, ferramentas e boas práticas que permitem o acesso e a análise de informação para melhorar e otimizar as decisões e o desempenho" de negócio [15].

2.1.1 Sistema de Business Intelligence

A Figura 2.1 ilustra o sistema de BI que inclui dois processos [28]: *getting data in* e *getting data out*.

Getting data in, processo de *data warehousing*, que envolve extrair os dados de vários sistemas operacionais da empresa e integrá-los num único repositório - o *Data Warehouse* (DW). Este processo e todos os seus componentes vão ser analisados, com mais pormenor, na Secção 2.2.

Getting data out, processo de *business intelligence*, que consiste num conjunto de aplicações disponibilizadas a todos os utilizadores de negócio (atuários, financeiros, técnicos ou comerciais) para aceder aos dados armazenados no DW e ajudá-los a tomar boas e rápidas decisões de negócio [28]. Estas aplicações devem fornecer as capacidades BI descritas na próxima secção, como por exemplo, *reporting* empresarial, análise OLAP e *data mining*.

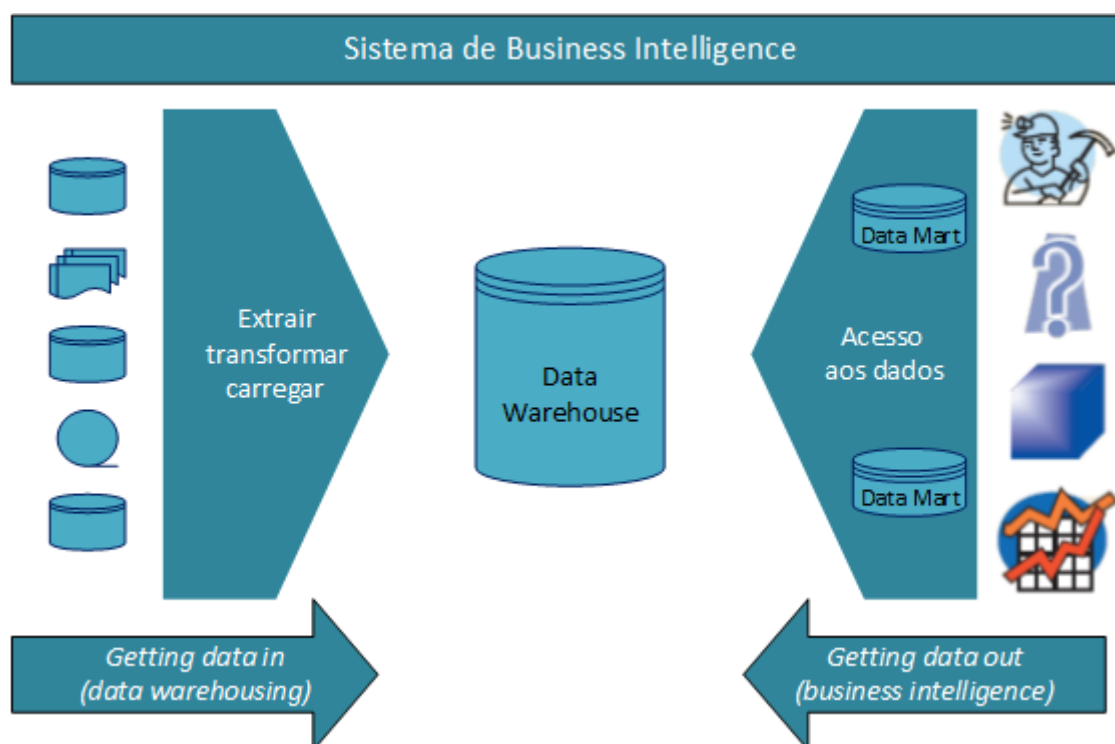


Figura 2.1: Sistema de Business Intelligence [28]

Neste projeto, o objetivo é implementar um sistema de *self-service* BI. *Self-service* BI visa permitir que a comunidade de utilizadores de negócios crie os seus próprios relatórios e análises a partir do zero, sem precisarem da ajuda da equipa de IT [12]. A vantagem do *self-service* BI é que a empresa, em particular a i2S, pode reduzir o número de colaboradores da equipa de BI que cria relatórios. Assim, a equipa de BI da i2S pode passar a prestar outros serviços como treinar os utilizadores, dar-lhes assistência ou a criar documentação e metadados, de forma a que os utilizadores consigam utilizar eficazmente as ferramentas e os dados disponibilizados.

Para ser possível criar um sistema de *self-service* BI, a aplicação disponibilizada aos utilizadores deve permitir que os utilizadores criem os seus próprios relatórios e explorem a informação no data warehouse sem precisarem de conhecer a sua estrutura. A interface gráfica desta aplicação deve ser fácil de utilizar e de perceber. Para conseguirmos obter um sistema de BI eficaz, a plataforma deve oferecer pelo menos as capacidades BI descritas na próxima secção.

2.1.2 Capacidades de Business Intelligence

Atualmente existe uma grande variedade de capacidades de BI que permitem às empresas tomar rápidas e melhores decisões. A MicroStrategy, que desenvolve software de Business Intelligence, agrupa esta variedade em cinco "estilos" de aplicações BI e disponibiliza cada um destes "estilos" através das ferramentas seguintes [17, 21]:

Análise OLAP

O OLAP (Online Analytical Processing) é uma tecnologia de análise dinâmica e multidimensional dos dados. A informação analisada por esta ferramenta OLAP é proveniente de base de dados multidimensionais (Data Warehouses) o que possibilita os utilizadores combinar a informação de muitas maneiras diferentes. Um utilizador pode analisar os dados utilizando as várias características da análise OLAP como:

- *Slice and dice*: restringe os dados que estão a ser analisados a um subconjunto desses dados;
- *Roll up (drill up)*: permite ao utilizador resumir a informação que está a analisar, ou seja, aumentar o grau de granularidade (i.e., granularidade mais grossa ou mais alta);
- *Drill down*: é o contrário do drill up e por isso, permite aumentar o nível de detalhe da informação que se está a visualizar, ou seja, diminuir o grau de granularidade (i.e., granularidade mais fina ou mais baixa);
- Ordenar a informação;
- Operações de *pivot*: mover um objeto das linhas para as colunas e vice-versa ou alterar a ordem dos objetos nas colunas ou nas linhas num relatório para ver os dados de diferentes perspetivas.

Relatórios empresariais (*Enterprise Reporting*)

Enterprise Reporting é uma tecnologia que permite às empresas criar relatórios com informação útil para ajudar na tomada de decisões. Estes relatórios fornecem informação mais detalhada que um *scorecard* ou um *dashboard*, definidos abaixo, e apesar de poderem ser ou incluir gráficos, normalmente apenas contêm texto.

Enterprise Reporting é o "estilo" BI mais predominante e é através destes relatórios empresariais que os utilizadores podem analisar a informação utilizando as características OLAP descritas no ponto anterior como o *drill up/down*.

No MicroStrategy consegue-se criar os três tipos de relatórios mais comuns:

1. **Relatórios operacionais**: contêm grandes quantidades de dados operacionais organizados num formato muito bem estruturado que são fundamentais para as operações de produção. Normalmente estes dados estão no nível de detalhe mais alto possível, ou seja, no nível transacional e são criados para suportar as atividades diárias de uma empresa. Por exemplo, um relatório que apresente o número de apólices em vigor, de apólices novas e de apólices anuladas por ramo, por ramo contabilístico e por ano.

2. **Faturas e declarações:** contêm dados transacionais detalhados e informações resumidas para qualquer número de clientes e parceiros.
3. **Relatórios de negócio:** apresentam informação da empresa de forma a ser possível avaliar o desempenho do negócio. Por exemplo, um relatório que compare a receita da produção (prémios emitidos e cobrados) com os sinistros (indenizações pagas e reservas constituídas).

Dashboards e Scorecards

Um *scorecard* é um tipo de relatório que mede o desempenho empresarial em relação a determinados objetivos usando indicadores chave de desempenho (em inglês, Key Performance Indicators (KPIs)). Num *scorecard*, cada indicador chave de desempenho representa um aspeto do desempenho empresarial e está sempre associado a um objetivo. Um indicador indica ao utilizador se os resultados estão longe ou perto do objetivo. De um modo geral, com estes KPIs, um *scorecard* mostra se uma empresa está a cumprir com sucesso ou a falhar um determinado plano estratégico.

Um *dashboard* permite juntar todo o tipo de relatórios, incluindo *scorecards*, num único documento e normalmente exibe apenas a informação mais importante recolhida desse conjunto de dados. Não são tão detalhados como os relatórios, mas podem dar respostas rápidas às perguntas mais importantes. É um painel de informação que oferece aos utilizadores um alto nível de interatividade na maneira como os dados são apresentados, permitindo juntar e combinar vários componentes (tabelas, gráficos, entre outros) e explorar/manipular os dados utilizando as várias capacidades OLAP como o *drill* e o *slice and dice*.

Data Mining e Análise Avançada

As empresas podem optar por uma grande variedade de métodos analíticos quando necessitam de algo mais complexo que os métodos básicos OLAP. São os métodos de análise avançada que podem ser baseados por exemplo em *data mining* ou análise preditiva.

Data mining é o processo de examinar grandes bases de dados à procura de padrões escondidos que possam representar informação útil, usando técnicas de *machine learning* e análise estatística. Com esta informação as empresas conseguem tomar melhores decisões de negócio. Existem dois tipos de tarefas de *data mining*: as descritivas, que nos dão informação e conhecimento sobre um conjunto de dados; e as preditivas que fazem previsões com base nos dados.

Aplicações móveis e serviço de alertas

Este último “estilo” tem duas capacidades principais. A primeira permite que as empresas construam uma ampla variedade de aplicações móveis essenciais para proporcionar Business Intelligence.

A segunda é o serviço que distribui relatórios personalizados e alertas por um grande número de utilizadores. Um exemplo simples deste serviço é o envio de emails programados (p.e. todas as segundas às 9 da manhã) para um conjunto de utilizadores com relatórios anexados. Outro exemplo, é o envio de um alerta caso seja detetado que o valor de um indicador de negócio saiu de um determinado intervalo aceitável.

2.1.3 Benefícios de Business Intelligence

Segundo Hugh Watson e Barbara Wixom, a utilização de um sistema de BI numa empresa, como, por exemplo, uma companhia de seguros, pode trazer os seguintes benefícios [28]:

- redução de custos com a infraestrutura de TI, ao eliminar os processos de extração de dados redundantes e a duplicação de dados armazenados em bases de dados independentes;
- poupança de tempo aos utilizadores e aos *data suppliers* porque o processo de entrega de dados é mais eficiente;
- acesso a mais e melhor informação;
- tomada de melhores decisões;
- melhoria dos processos de negócio através de análises preditivas;
- suporte para a execução dos objetivos estratégicos.

Estes benefícios são atingidos ao longo do tempo. À medida que os utilizadores de negócio começam a ficar mais experientes a realizar análises, o nível dos benefícios torna-se mais global e difícil de quantificar.

2.2 Data Warehouse

Um *data warehouse* é um repositório de informação atual ou histórica de uma empresa, obtida a partir de vários sistemas operacionais, que serve para facilitar a análise dos grandes conjuntos de dados normalmente existentes nas empresas e a consequente tomada de decisões de negócio.

Os data warehouses são necessários devido à dificuldade existente em analisar os dados diretamente nesses sistemas OLTP (Online Transaction Processing/ Processamento de Transações em Tempo Real). Estes sistemas armazenam os dados relativos à atividade de uma empresa e o problema é que costumam conter grandes quantidades de dados detalhados e esta informação costuma estar desorganizada e dispersa por vários sistemas. Isto leva a problemas de desempenho nas consultas e execução de relatórios, bem como a possíveis problemas na integração desta informação, impossibilitando uma análise eficiente. Estes sistemas também não estão preparados para serem consultados de uma forma geral e de maneiras inesperadas por parte dos utilizadores, já que têm de estar sempre disponíveis para registar as transações de negócio da empresa e não a processar consultas.

Como vamos ver na Subsecção 2.2.2, um processo de data warehousing não se traduz apenas neste armazenamento de informação, mas também num processo de ETL (Extract, Transform and Load). Só depois deste processo é que são disponibilizados dados proveitosos para os utilizadores poderem fazer consultas complexas sobre o negócio de uma empresa, através de ferramentas de Business Intelligence.

Existem várias metodologias que podem ser adotadas no desenvolvimento de um data warehouse. As duas abordagens mais relevantes são a de Inmon e a de Kimball [5].

A abordagem sugerida por Inmon chama-se *top-down*. Nesta metodologia, o DW é o elemento central de todo o ambiente analítico [9]. Um data warehouse contém dados transacionais ou altamente detalhados que são extraídos de vários sistemas operacionais e integrados num modelo de dados empresarial normalizado.

Em [18], Inmon define um data warehouse como sendo uma coleção de dados que apoia as decisões de gestão e que é:

- Orientado por assuntos: nos sistemas operacionais, os dados estão organizados por aplicações (p.e., nas companhias de seguros, as aplicações são vida, não-vida, etc). O data warehouse deve organizar esses dados por assunto/tema (p.e., sinistro, apólice, prémio, cliente, etc).
- Integrado: é atribuída uma representação única e universal a todos os dados provenientes dos vários sistemas operacionais. Normalmente, os dados nos sistemas operacionais têm formatos diferentes, ou seja, diferentes convenções de nomenclatura, codificação, etc, e assim, quando são carregados no data warehouse essas inconsistências são desfeitas.
- Não volátil: os dados no data warehouse nunca são atualizados (no sentido geral), só inseridos e consultados. Quando existem atualizações nos sistemas operacionais, são inseridos novos dados no DW. O objetivo é manter um histórico dos dados.
- Variante no tempo: os registos estão marcados no tempo (p.e., têm a data da transação) para ser possível fazer análises ao longo do tempo, como previsões e comparações.

A abordagem de Kimball chama-se *bottom-up* e nesta abordagem o data warehouse pode ser descentralizado. Vários subconjuntos de dados podem estar armazenados em diferentes máquinas e base de dados (BD), e controlados por diferentes departamentos autónomos, mas ligados através de uma arquitetura que faz com que eles possam ser combinados de forma eficaz. Esses subconjuntos de dados são chamados de *data marts*, conceito descrito na Subsecção 2.2.1. Estes data marts contêm todos os dados - detalhados e sumarizados.

Segundo Ralph Kimball, um data warehouse tem os objetivos seguintes [20]:

- Colocar a informação da empresa facilmente acessível: o conteúdo do data warehouse deve ser compreensível e as ferramentas que acedem ao DW devem ser simples e fáceis de utilizar.
- Apresentar a informação de forma consistente: assegurar a qualidade e credibilidade da informação. Informação extraída de uma parte da empresa deve ser combinada com a informação extraída de outra parte. Por exemplo, se dois campos de duas tabelas diferentes tiverem o mesmo nome, então têm que significar a mesma coisa. Se não tiverem o mesmo significado, então têm que ter nomes diferentes.
- Ser adaptável e resistente à mudança: o DW deve ser concebido para conseguir lidar com mudanças inevitáveis, tais como, novas necessidades dos utilizadores, condições de negócio, dados ou tecnologias. Os dados e aplicações existentes não devem ser alterados ou interrompidos quando os utilizadores solicitam novas questões ou quando são inseridos novos dados no DW.
- Ser seguro de forma a proteger a informação confidencial da empresa.
- Servir de base para melhorar as tomadas de decisões.
- Ser aceite e utilizado ativamente pela comunidade de negócios.

Para este projeto, adotei a abordagem de Kimball e os conceitos descritos nas próximas secções são sobre esta metodologia. Na (Subsecção 2.2.1) vou explicar porque foi adotada esta abordagem e não a de Inmon.

Estas metodologias de desenvolvimento e as diferentes arquiteturas de DW existentes vão ser descritas com mais detalhe na Secção 4.3.

2.2.1 Data Mart

Um data mart (DM) é um subconjunto lógico de um data warehouse. Um data warehouse é composto da união de todos os seus data marts. Cada um desses data marts pode conter um conjunto de dados relativos a um único processo de negócio ou relativos a um grupo de

processos de negócio de um determinado setor de negócio. Um processo de negócio é uma atividade realizada na empresa que é suportada por um sistema de origem de recolha de dados e pode abranger as encomendas ou as faturas, ou no caso das companhias de seguros, as apólices ou os sinistros. Normalmente um data mart é criado de forma a ter apenas os dados essenciais de um único processo de negócio e ser assim menos confuso que um data warehouse [19].

Podemos implementar os data marts de duas maneiras. Podemos começar por criar um data warehouse centralizado que contenha todos os dados juntos provenientes de vários sítios da empresa e só depois implementar vários data marts dependentes como subconjuntos desse data warehouse (abordagem *top-down*). Ou então podemos criar logo um data mart independente para cada processo de negócio e, só quando for necessário, integrar todos os data marts de forma a termos um data warehouse mais abrangente (abordagem *bottom-up*).

Os data marts dependentes têm a vantagem de se conseguir criar um modelo de dados mais consistente e ser mais fácil de garantir a qualidade dos dados, já que os dados são todos extraídos e transformados pelo mesmo processo de ETL e organizados e armazenados no mesmo data warehouse. No entanto, a construção de um data warehouse centralizado e completo antes da implementação dos data marts e consequentemente das aplicações e relatórios, não é a melhor solução em empresas de grandes dimensões que estão em constante mudança e a obter novas fontes de dados.

A solução mais realista é a construção de data marts independentes, porque as empresas só precisam de preparar para consulta um pequeno conjunto de dados relativos a um setor ou processo de negócio e já podem começar a desenvolver as suas aplicações e relatórios. Desta forma, poupam muito mais tempo e dinheiro [20, 9]. Na abordagem *bottom-up*, um projeto pode representar um único DM que pode ser implementado rapidamente, enquanto na abordagem *top-down*, um projeto pode-se tornar num enorme empreendimento impossível [19]. Logo, devido a esta razão, optou-se pela abordagem de Kimball para este projeto.

No entanto, esta abordagem não é preferível se os data marts independentes estiverem isolados e não puderem ser ligados entre eles. Para evitar isto deve-se aderir ao que Ralph Kimball chama de arquitetura de data warehouse em *bus* (DWB), ou em série [20]. Esta adesão à arquitetura DWB, consiste em garantir que todos os data marts tenham dimensões e factos comuns entre eles em conformidade, ou seja, tenham dimensões que possam ser partilhadas entre factos em vários data marts. São estas dimensões que vão permitir a integração dos data marts. O resultado vão ser muitas tabelas de factos de diferentes data marts a partilhar as mesmas tabelas de dimensões. Com isto os utilizadores vão poder analisar informação integrada de vários data marts e só assim se pode evitar a estrutura centralizada do DW. Na Secção 5.4, vou descrever este assunto com mais detalhe.

Os data marts independentes também não devem conter apenas dados agregados e servirem apenas para responder a um conjunto de questões de negócio feitas por um conjunto de utilizadores específico ou por um dado departamento na empresa (abordagem *top-down*). Os dados

presentes devem estar no nível mais baixo de granularidade, de forma a suportar consultas ad hoc por parte dos utilizadores [20].

2.2.2 Processo de Data Warehousing

Existem vários componentes no processo de data warehousing e o primeiro deles são os sistemas OLTP. A informação presente no data warehouse deriva destes dados e como já vimos anteriormente, é importante que o data warehouse esteja separado destes sistemas.

Entre os sistemas OLTP e o data warehouse existe ainda uma área de estágio. Esta área também é um local de armazenamento e é aqui onde é executado o processo ETL, descrito na próxima secção, que termina com o carregamento dos dados no data warehouse.

O último componente do processo de data warehousing, é o próprio data warehouse, onde os dados são organizados, armazenados e disponibilizados para os utilizadores poderem consultá-los e analisá-los.

2.2.3 Processo ETL

Uma das operações mais decisivas e que consomem mais tempo no processo de data warehousing é a extração, transformação e carregamento de dados provenientes de vários sistemas num data mart ou num data warehouse, chamada ETL (Extract, Transform and Load). O processo ETL consiste em três tarefas [20]:

- Extração: A primeira tarefa a ser executada é a extração de todos os dados necessários dos sistemas de origem.
- Transformação: Depois de extraídos, é preciso assegurar a qualidade dos dados antes de carregá-los no data mart ou no data warehouse. Existem várias transformações possíveis, tais como combinar dados provenientes de múltiplas fontes ou limpar os dados, como por exemplo, corrigir simples erros nos valores, remover as linhas duplicadas nas tabelas, garantir que os valores de um mesmo objeto estão no mesmo formato ou garantir a consistência entre os objetos. Outras operações que podem ser realizadas nesta fase são a remoção dos dados que não interessam passar para o DW e a criação de tabelas de agregadas.
- Carregamento: A última tarefa consiste em carregar os dados no data warehouse.

Num processo ETL podemos armazenar os dados que foram extraídos numa área física ou podemos realizar as transformações diretamente em memória.

Na primeira opção é utilizada uma área de estágio situada entre os sistemas de origem e o data warehouse. Esta área de armazenamento pode ser, por exemplo, uma base de dados relacional ou um sistema de *flat files*. Uma das vantagens da utilização desta área é que, caso haja uma falha numa transformação, podemos recuperar sem termos de ir buscar novamente os dados ao sistema de origem. Também evitamos ter que voltar a realizar a transformação dos dados se o processo de carregamento falhar. Outra das vantagens é a possibilidade de fazermos backups do data warehouse já que temos as tabelas ou os ficheiros da área de estágio que nos permitem voltar a recarregar o data warehouse. A última grande vantagem é o facto de podermos verificar se não ocorreu nenhuma falha no processo ETL já que podemos manter as tabelas originais que foram extraídas dos sistemas de origem na área de estágio. Assim se houver alguma dúvida em relação à integridade dos dados no data warehouse, podemos utilizar as tabelas originais para comparar. Isto é especialmente útil quando o histórico dos sistemas de origem é alterado e não podemos recuperar as tabelas originais. Por fim, os utilizadores finais e as aplicações BI não devem fazer consultas nesta área porque podem ter problemas de desempenho e de qualidade dos dados. Esta é somente uma área para realizar transformações nos dados e por isso não está pronta para ser consultada [20].

Se optarmos por não utilizar uma área de estágio e preferirmos realizar as transformações em memória, então o processo vai ser mais rápido. No entanto, com esta opção não vamos ter as vantagens que teríamos se utilizássemos uma área de estágio e por isso se quisermos criar um processo eficiente devemos encontrar o equilíbrio ideal entre guardar os dados em tabelas ou mantê-los em memória [20]. Por exemplo, se tivermos um pequeno conjunto de dados que necessitem apenas de umas simples transformações, então podemos optar por realizá-las em memória para tornar o processo mais rápido. Contudo, esta opção depende sempre da quantidade de memória que o servidor de ETL tem.

2.2.4 Modelo Dimensional

O modelo dimensional é uma técnica utilizada atualmente no desenho dos data warehouses. Segundo Ralph Kimball, o modelo dimensional é a técnica mais viável para entregar dados aos utilizadores do DW porque apresenta os dados de uma forma simples e intuitiva, além de permitir um elevado desempenho nas consultas [20].

Problemas do modelo na terceira forma normal

O modelo dimensional é diferente do modelo na terceira forma normal (3FN). O modelo na 3FN é uma técnica usada para eliminar as redundâncias de dados, para garantir a consistência dos dados e para poupar espaço de armazenamento. É um modelo mais indicado para as bases de dados operacionais, porque nelas existe uma maior independência entre os dados e por isso, não costuma haver anomalias nas operações de inserir ou atualizar uma transação. No

entanto, como esta técnica utiliza imensas tabelas de forma a separar os dados de assuntos diferentes, algumas consultas nestas bases de dados normalizadas podem ser bastante lentas se for necessário fazer várias junções entre tabelas. Além disso, devido ao elevado número de tabelas, estes modelos são muito complexos e difíceis de perceber por parte dos utilizadores. Assim, este modelo não é o mais indicado para os data warehouses [20].

ROLAP vs MOLAP

O modelo dimensional tanto pode ser aplicado em bases de dados relacionais como multidimensionais. As multidimensionais são bases de dados criadas especialmente para data warehouses e para aplicações OLAP. Chama-se MOLAP (Multidimensional Online Analytical Processing) às ferramentas OLAP que acedem a dados armazenados em bases de dados multidimensionais [26]. Nestas bases de dados, a informação é armazenada em cubos dimensionais. Estes cubos já têm todas as combinações possíveis entre os valores dos atributos, em que cada uma delas pode ser acedida diretamente [25]. Cada um dos cubos contém todas as respostas possíveis para um determinado conjunto de questões e, por isso, as suas principais vantagens são o alto desempenho nas consultas e o facto de permitir cálculos complexos. A baixa escalabilidade é a principal desvantagem, já que os cubos não conseguem armazenar e manipular grandes quantidades de dados [13], sendo a solução utilizar apenas informação de nível de resumo e não de detalhe.

Se os dados forem armazenados numa base de dados relacional então o modelo dimensional é implementado usando um esquema em estrela (*star schema*), como ilustrado na Figura 2.2. Um esquema em estrela é composto por uma tabela de factos rodeada por um conjunto de tabelas de dimensões formando assim uma estrela. Mais à frente vamos descrever o que são estas tabelas de factos e dimensões. Este tipo de esquema (modelo dimensional) é mais desnormalizado que o modelo 3NF e assim permite um maior desempenho nas consultas porque não é necessário realizar tantas e complexas junções entre tabelas [20].

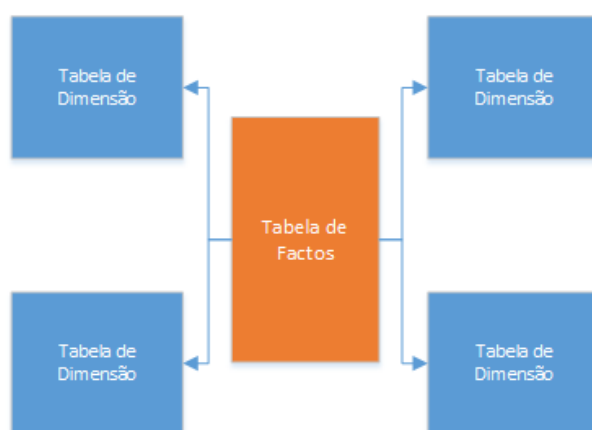


Figura 2.2: Esquema em Estrela

Este esquema também pode ser em floco de neve (*snowflake*), como ilustrado na Figura 2.3, que é uma variação do esquema em estrela, em que as tabelas de dimensão estão normalizadas em múltiplas tabelas de dimensão relacionadas entre si, formando assim um floco de neve. No entanto, as tabelas de dimensão devem ser desnormalizadas, visto que o tamanho das tabelas de dimensão é sempre muito menor que o tamanho das tabelas de factos e, por isso, não há grandes vantagens em normalizar estas tabelas para tentar poupar espaço em disco porque o impacto no tamanho total da BD vai ser muito reduzido [20]. Assim, devemos evitar o esquema em floco de neve, porque pode trazer problemas de desempenho nas consultas, já que o número de operações de junção de tabelas necessárias pode ser maior. Contudo, existem algumas situações em que podemos considerar este esquema. Na Secção 6.6 vamos descrever uma dessas situações.

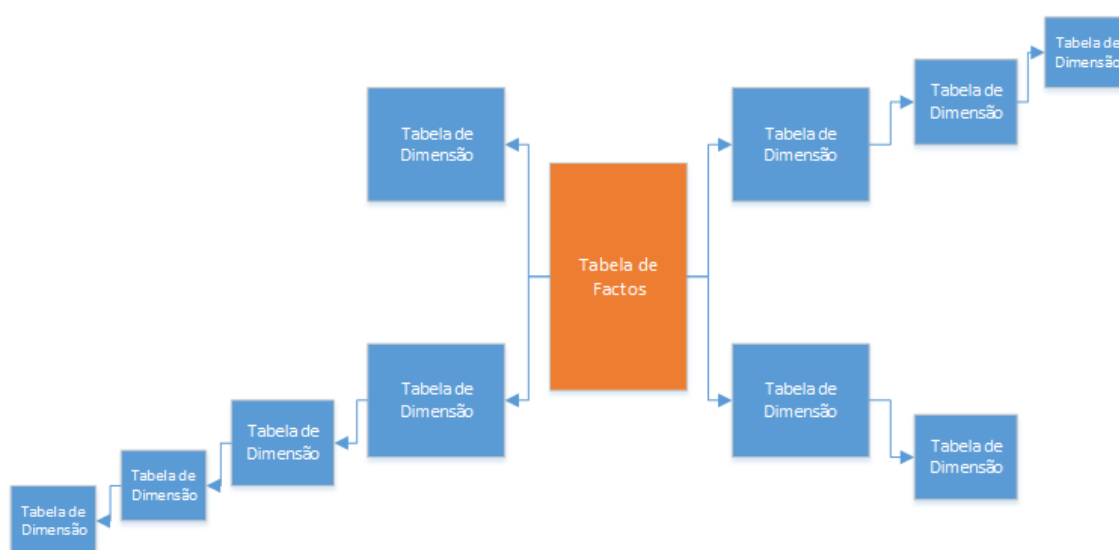


Figura 2.3: Esquema em Floco de Neve

As aplicações OLAP que acedem a dados armazenados em bases de dados relacionais chamam-se ROLAP (*Relational Online Analytical Processing*). As bases de dados relacionais são mais adequadas para guardar grandes quantidades de dados que as bases de dados multidimensionais, tendo alta escalabilidade. A desvantagem é que a tecnologia ROLAP é obrigada a utilizar as linguagens de manipulação de dados dos sistemas de bases de dados relacionais para realizar operações, como por exemplo a linguagem SQL [6]. Isto é um problema porque estas linguagens não são apropriadas para operações multidimensionais, levando a um baixo desempenho e à incapacidade de realizar cálculos complexos.

Existem aplicações OLAP que combinam as vantagens do MOLAP e ROLAP que chamam-se HOLAP (*Hybrid Online Analytical Processing*). Neste caso, a informação sumarizada é armazenada em cubos OLAP para se obter um maior desempenho. Quando é necessário informação detalhada, estas aplicações fazem o *drill* para a base de dados relacional.

Neste projeto, vamos utilizar uma base de dados relacional e o sistema de base de dados

utilizado para o DW é o SQL Server 2008. Optou-se por uma base de dados relacional porque a plataforma MicroStrategy tem uma tecnologia, descrita na Secção 3.1.4, que combate as desvantagens do ROLAP.

Tabela de factos

Uma tabela de factos é a tabela primária num modelo dimensional, onde são armazenadas as medidas dos processos de negócio. Um facto é uma coluna numa tabela de factos e normalmente contém dados numéricos e aditivos. Exemplos de factos: prémio da apólice, valor do sinistro e número de pessoas seguras. Estas tabelas costumam ter muitas linhas e, por isso, deve-se ter em atenção o espaço em disco que ocupam.

Além dos factos, estas tabelas também são compostas por *foreign keys* (chaves estrangeiras/secundárias) que ligam às chaves primárias das tabelas de dimensão. A chave primária da tabela de factos acaba por ser uma combinação das *foreign keys*.

Existem três tipos de tabelas de factos [20]:

- Tabela transacional: é a mais comum das três e é a tabela que regista uma linha por cada transação, ou seja, uma linha é criada sempre que ocorre um evento. Normalmente são as tabelas que contêm dados do nível mais detalhado e por isso que ocupam mais espaço em disco;
- Tabela de sumarização periódica: é usada para mostrar a atividade de negócio ocorrida durante intervalos de tempo regulares. Pode mostrar medidas instantâneas no fim de um período ou medidas acumuladas durante o período.
- Tabela de sumarização acumulada: Estas tabelas armazenam todas as fases de um evento numa só linha e têm várias dimensões de datas. Assim, conseguimos perceber qual é o estado atual de um dado processo a qualquer altura. Por exemplo, podemos ter uma tabela que registe numa só linha todas as fases de cada cobertura de cada apólice, desde que ela é registada até deixar de estar em vigor. Neste tipo de tabelas quando existe uma alteração num dado evento, a linha é atualizada em vez de ser criada uma nova linha. Nestas tabelas é normal haver factos com valor nulo. Por exemplo, se para cada cobertura houver um campo que indique a data para o primeiro sinistro existente, se ainda não tiver havido nenhum sinistro é normal que tanto esta dimensão como os factos associados não tenham valores. Estas tabelas são mais pequenas comparadas com as outras e têm muitas mais dimensões.

Existem três tipos de factos [20]:

- Aditivos: factos que podem ser somados ao longo de todas as dimensões;

- Semi-aditivos: factos que só podem ser somados ao longo de algumas das dimensões;
- Não-aditivos: factos que não podem ser somados (por exemplo: taxa de sinistralidade).

Tabela de dimensão

As tabelas de dimensão armazenam atributos que servem para descrever os factos. Por exemplo, podemos considerar o facto "indenizações pagas" por uma companhia de seguros. Se um relatório apenas nos mostrar que o valor desse facto é de 15000, este valor por si só dá-nos pouca informação. Mas, se esse relatório contiver as indenizações pagas por data, por titular e por ramo, já vamos obter um contexto. Assim, os atributos são utilizados para responder a questões sobre o negócio acerca de factos em vários níveis de detalhe [22]. Uma curiosidade, é que numa consulta ou num pedido de relatório, os atributos são especificados pelas palavras "por" (*by*) [20]. Cada tabela de dimensão tem uma só chave primária que serve para ligar a uma ou mais tabelas de factos.

Capítulo 3

Ferramentas Utilizadas no Projeto

3.1 Plataformas de Business Intelligence

As plataformas de BI são um conjunto de ferramentas que ajudam os utilizadores a consultar e a analisar os dados presentes no data warehouse, transformando esses dados em informação útil de negócio. Existem várias plataformas no mercado que oferecem grande parte das capacidades de BI existentes, como por exemplo, o Oracle Business Intelligence, o Microsoft Business Intelligence, o QlikView da QlikTech ou o MicroStrategy.

3.1.1 MicroStrategy

O *software* da MicroStrategy foi a plataforma de Business Intelligence escolhida pela i2S para permitir que as companhias de seguros consigam tomar decisões de negócio com base nos dados presentes no data warehouse.

3.1.2 Porquê MicroStrategy?

Atualmente existem várias aplicações de Business Intelligence e por isso nem sempre é fácil escolher a que mais se ajusta ao nosso projeto. Para facilitar a escolha podemos recorrer ao Quadrante Mágico para plataformas analíticas e de BI da Gartner Group [16], empresa de pesquisa e consultoria na área de tecnologia da informação, que mostra os principais fornecedores deste software. Estes fornecedores estão distribuídos por quatro quadrantes que determinam a sua classificação anual como líderes, desafiadores, visionários ou como pertencentes a um nicho de mercado. Analisando o Quadrante Mágico na Figura 3.1 consegue-se descobrir que em Fevereiro de 2013 apenas dez das empresas eram líderes de mercado e, por isso, seriam as que à partida ofereciam as melhores soluções. Entre estas dez, a i2S optou pelo MicroStrategy,

devido às razões descritas mais à frente.

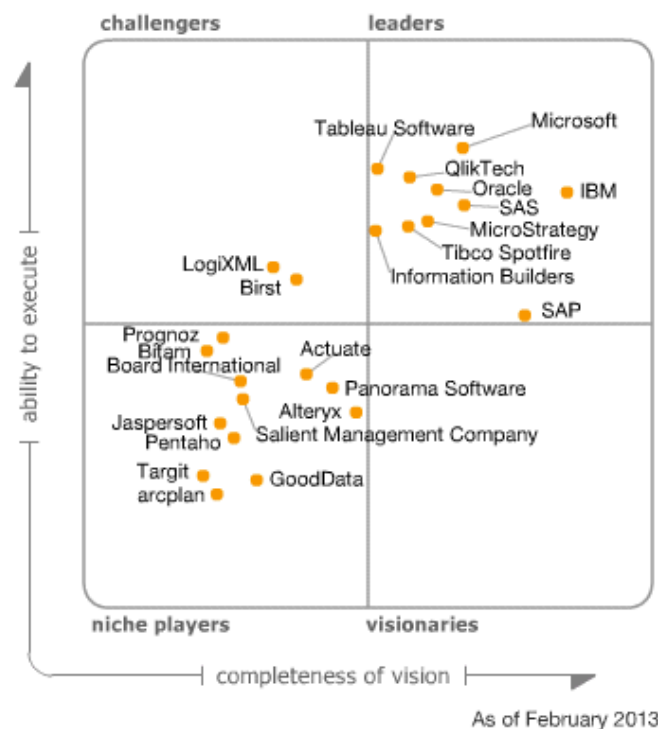


Figura 3.1: Quadrante Mágico para plataformas analíticas e de BI da Gartner Group [16]

Existem dois tipos principais de ferramentas de BI: as de Data Discovery (DD) e as tradicionais.

As tradicionais são ferramentas BI que permitem aos utilizadores seleccionar o conjunto de dados que querem ver, construindo consultas através de *drag and drop*. Depois de definido este conjunto de dados, pode-se construir facilmente um relatório ou dashboard nestas ferramentas. No entanto, apesar de ser fácil a implementação de dashboards para as equipas de TI e para os próprios utilizadores de negócio, estes dashboards só oferecem algumas capacidades de análise *ad-hoc* e são formatos muito fixos que só permitem tomar decisões reativas. Estas ferramentas não são projetadas para inspecionar grandes volumes de dados e limitam as combinações de informação que os utilizadores podem realizar quando estão a ver estes relatórios.

Assim, houve a necessidade de criar ferramentas de DD que oferecem uma maior flexibilidade nas análises. Permitem aos utilizadores uma maior autonomia no acesso à informação proveniente de várias fontes de dados, maior facilidade em explorar grandes quantidades de dados, em observá-los de diferentes formas e encontrar respostas. Normalmente estas ferramentas, têm arquiteturas *in-memory* e interfaces mais gráficas e interactivas, sendo mais fácil para os utilizadores explorar os dados através de operações como o *drill* e experimentar diferentes tipos de apresentações como mapas, *heatmaps* (matriz em que os seus valores individuais são representados por cores) ou animações de séries temporais (*time series*). Estas ferramentas permitem aos utilizadores a livre exploração e descobrir combinações e padrões nos dados sem

terem de modelar as relações entre esses dados previamente.

Apresentamos a seguir as conclusões de [16]:

A QlikTech, fundada na Suécia, foi pioneira e é líder de mercado em Data Discovery, com o seu produto BI chamado QlikView. QlikView é uma plataforma baseada num armazenamento de dados *in-memory* e em *direct query access* (atualmente tem o Teradata, Google BigQuery e Cloudera), com um conjunto de ferramentas de BI bem integradas.

Segundo o estudo/inquéritos da Gartner, as principais razões que levam as pessoas a escolherem o QlikView são: a facilidade de utilização para os utilizadores finais, o grande número de funcionalidades e o alto desempenho devido ao modelo computacional *in-memory* da plataforma. Nos inquéritos, o QlikView teve classificações positivas em relação ao *reporting*, análises *ad hoc*, dashboards, visualizações interativas, scorecards, *search-based data discovery* e *mobile BI*. Como é um fornecedor de DD, os seus utilizadores também reportaram benefícios no negócio acima da média, particularmente ao disponibilizar melhores informações para mais utilizadores e ao expandir o tipo de análises realizadas. O QlikView tem uma das menores percentagens de clientes que planeiam deixar de usar o produto no futuro e estão entre aqueles que mais sucesso tiveram nas classificações.

O maior problema da QlikTech é que DD, apesar de ter tido um grande crescimento, está a tornar-se *mainstream*. O mercado está a ficar muito competitivo, com novos fornecedores a emergirem e com outros fornecedores, que também lideram o mercado de BI, a introduzir capacidades de DD nas suas plataformas. Outros problemas da aplicação são as funcionalidades empresariais, como a gestão dos metadados, infraestrutura BI e ferramentas de desenvolvimento BI que tiveram baixa classificação por parte dos clientes. Isto leva a que os utilizadores demorem mais tempo a desenvolver relatórios, quando comparado com muitas outras plataformas. Também a gestão da segurança e a administração para grandes números de utilizadores tiveram baixa pontuação. A falta de escalabilidade no QlikView quando são incluídos mais utilizadores, maiores quantidades de dados e desenvolvidos dashboards mais complexos, também é outro problema. O limite de tamanho de dados é devido à opção da tecnologia *in-memory*, enquanto os seus concorrentes podem consultar diretamente e trazer dados dos DWs. Por fim, o software é caro comparado com outras alternativas.

Como referi anteriormente, a concorrência da QlikView e das outras empresas de DD está a aumentar. As empresas que antes só desenvolviam ferramentas de BI tradicionais, atualmente também estão a desenvolver ferramentas de DD. Por exemplo, a IBM com o Cognos Insight e a MicroStrategy com o Visual Insight.

A IBM lidera no eixo que classifica as empresas como visionárias no Quadrante Mágico. Entre muitos aspetos, um dos pontos fortes da IBM é o facto de oferecer um conjunto de funcionalidades BI a um baixo preço, com uma limitação até 100 utilizadores finais, conseguindo assim atrair as empresas/departamentos de menor dimensão. Segundo os inquéritos da Gartner,

as principais razões que levam as pessoas a escolher a plataforma da IBM são as tecnologias *road map* (que permitem definir objetivos de negócio e ajudar a cumpri-los) e as que permitem prever o futuro, a qualidade do produto e a capacidade de integração com infraestruturas de informação (base de dados, middleware).

O principal problema desta plataforma é o baixo desempenho. No entanto, a IBM já está a planear inserir novas tecnologias no seu produto de forma a combater este ponto negativo, como a tecnologia *in-memory* ROLAP. Outros pontos abaixo da média são a dificuldade de utilização por parte dos utilizadores finais e o tempo médio que se demora a fazer relatórios.

A plataforma MicroStrategy (MSTR), escolhida para este projeto, tem alguns pontos fortes. É um dos produtos que tem mais funcionalidades e que tem mais capacidade para grandes volumes de dados e para muitos utilizadores. Estes foram alguns dos pontos que levaram a i2S a optar por esta plataforma BI.

Todas as ferramentas desta plataforma foram desenvolvidas pela empresa a partir do zero. Não houve integração de produtos através da aquisição de fornecedores mais pequenos. Por isso, os componentes desta plataforma têm um grande nível de integração. Segundo a Gartner, por causa disto os clientes identificam a qualidade no produto (estabilidade, confiabilidade e correção) como uma das principais razões para a sua escolha.

Outro ponto forte são os seus produtos Mobile BI e Cloud BI. O Mobile BI é compatível com os aparelhos BlackBerry, Apple e Android. A empresa lidera no facto de 50% dos seus clientes inquiridos usarem ativamente a mobilidade para BI. Também é um dos líderes em termos de qualidade do conjunto das capacidades móveis, sendo que em 2012 era mesmo o líder. Como a i2S e os seus clientes também estão muito interessados em *mobile* BI, isto também pesou muito na hora de escolher.

Também o produto Visual Insight tem tido um forte desenvolvimento, com novas capacidades de visualização, mais usabilidade, mais funções analíticas e melhor integração nos produtos *mobile* e *cloud*.

Em relação aos pontos fracos, a plataforma MSTR está abaixo da média no que toca à facilidade de utilizar para a equipa de desenvolvimento e para os utilizadores finais. Mobile BI e o Visual Insight irão com certeza ajudar a melhorar a usabilidade mas atualmente continuam abaixo da média. Outra desvantagem é o custo acima da média nas licenças e implementações. Por fim, outro dos problemas é o facto das capacidades analíticas não serem o forte da MSTR e, por isso, está mal classificada no eixo dos Visionários. A plataforma oferece capacidades nativas de mineração de dados (*data mining*), mas estas continuam a ser ignoradas pelos utilizadores. A MSTR tem a análise preditiva menos usada de todos os fornecedores do Quadrante Mágico. A razão por trás disto pode ter que ver com a interface do utilizador que foca mais no desenho de relatórios que nas capacidades de *data mining*.

Para concluir, foi escolhida a MSTR, mas qualquer uma das outras plataformas das dez empre-

sas líderes de mercado seria uma boa opção. E, se no futuro, uma companhia de seguros que seja cliente da i2S preferir outra destas aplicações, ela pode ser implementada sobre o DW que foi desenvolvido no âmbito deste trabalho de estágio.

3.1.3 Componentes da Plataforma MicroStrategy (MSTR)

A plataforma da MSTR oferece um conjunto de componentes para a criação e desenvolvimento de soluções de BI [22]. Alguns dos principais componentes da plataforma são:

MSTR Architect: é uma ferramenta de desenvolvimento que nos permite definir o modelo lógico dos dados, mapeando este modelo lógico com as tabelas físicas do data warehouse. É aqui que podemos criar os objetos do esquema (*schema objects*) que são os factos, atributos e hierarquias.

MSTR Desktop: é um componente que nos dá acesso aos cinco "estilos" de Business Intelligence a partir de uma única interface. É no MSTR Desktop que podemos criar os projetos de BI. É também neste componente que podemos criar e gerir objetos da aplicação (*application objects*) para cada projeto tais como filtros, prompts, métricas e relatórios. Mas, antes de podermos criar estes *application objects*, temos que ter os *schema objects* necessários já definidos, porque são eles que permitem o acesso aos dados para análise presentes no DW. Por exemplo, os factos são usados para criar métricas que por sua vez são usadas para desenhar relatórios. Os *schema objects* também podem ser criados no Desktop. No fim, estes objetos vão servir os utilizadores do MSTR Desktop e do MSTR Web.

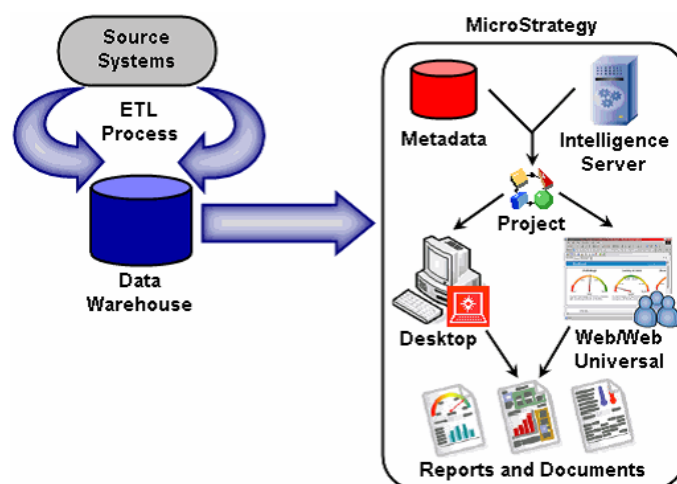


Figura 3.2: Componentes da plataforma MicroStrategy [22]

MSTR Project: Os projetos servem para podermos agrupar os dados do data warehouse e os objetos que são depois criados na aplicação, em pequenos conjuntos destinados a certos grupos de utilizadores. Assim, os relatórios e outros objetos não são misturados e um grupo de

utilizadores vê apenas aquilo que lhe interessa e que é da sua área de análise. Cada projeto tem o seu repositório de metadados e as suas tabelas do DW específicas.

MSTR Web: tem uma interface mais simples e intuitiva para os utilizadores poderem fazer *reporting* e análises. Pode-se aceder a esta interface *web* na maioria dos *web browsers*. O MSTR Web pode ser instalado na maioria dos servidores *web* (Apache HTTP Server, Microsoft IIS Windows Server, IBM HTTP Server, etc.). Como o MSTR Web é mais direcionado para os utilizadores do que para os implementadores, este é mais limitado que o MSTR Desktop em relação a funcionalidades.

MSTR Intelligence Server: realiza o processamento analítico e a gestão de tarefas para todas as aplicações do MSTR. Funciona como a componente central que conecta o MSTR Desktop, os metadados, o DW e o *web server*. Partilha os objetos e a informação entre esses componentes, gere essa partilha e protege a informação presente nos metadados.

MSTR metadata: é um repositório que armazena as definições de todos os objetos criados no MSTR, como os relatórios, métricas, *templates*, hierarquias, atributos, objetos de configuração, etc. Os metadados são dados que descrevem outro conjunto de dados e servem para mapear os objetos presentes no MSTR com os dados do DW. Este repositório pode estar na máquina que aloja o data warehouse ou noutra máquina qualquer.

3.1.4 In-memory ROLAP

A plataforma BI da MicroStrategy está mais vocacionada para ROLAP do que para MOLAP, mas também pode atuar sobre bases de dados multidimensionais. A grande novidade nesta ferramenta é a utilização de uma tecnologia chamada *In-memory* ROLAP para combater a grande desvantagem das aplicações ROLAP que é o baixo desempenho nas consultas [21].

Esta tecnologia permite guardar conjuntos de dados retornados do data warehouse em cubos dimensionais na memória do Intelligence Server. Estes cubos dimensionais que residem em memória chamam-se Cubos Inteligentes (*Intelligent Cubes*) e as suas definições são armazenadas no repositório de metadados. Basicamente com estes cubos é implementada uma estrutura multidimensional em cache.

Depois de implementados os cubos, os utilizadores podem criar relatórios utilizando estes dados carregados em memória. Os dados presentes nestes Cubos Inteligentes podem ser partilhados entre vários relatórios criados por múltiplos utilizadores. O objetivo desta tecnologia é então diminuir o tempo de execução dos relatórios, já que a consulta é previamente executada e os resultados armazenados no cubo. No entanto, se utilizarmos intensivamente esta solução devemos ter cuidado com a quantidade de RAM disponível para armazenar os cubos [23].

3.2 Ferramentas de ETL

Muitas empresas têm processos de data warehousing complexos e dispendiosos, que consomem muito tempo no seu desenvolvimento e manutenção. Por isso, com o propósito de reduzir o esforço necessário para implementar e gerir estes processos, foram desenvolvidos vários tipos de ferramentas de ETL, desde as mais simples que apenas fazem a migração dos dados até às mais avançadas que realizam o processo de ETL completo.

Neste projeto foi utilizada uma ferramenta de ETL, visto que criar manualmente vários *stored procedures* ou *scripts* para integrar os dados no data warehouse iria dar mais trabalho e iria demorar mais tempo a implementar. Com o processo manual temos um maior controlo sobre o processo e as possibilidades de customização são numerosas. No entanto, as ferramentas de ETL trazem vantagens mais significativas, tais como:

- maior facilidade e rapidez em desenvolver uma tarefa de integração de dados, já que normalmente não é necessário programar porque a maior parte das funções necessárias já estão criadas;
- escalabilidade, pois podemos transferir uma ferramenta de ETL para outro servidor mais facilmente ou então distribuir as tarefas entre vários servidores;
- maior desempenho, pois o processo de ETL desenvolvido nestas ferramentas normalmente consome menos recursos e é realizado com maior velocidade. Claro que é possível construir um sistema de ETL com bom desempenho se não usarmos uma ferramenta. No entanto, a estrutura imposta por uma ferramenta de ETL faz com que seja mais fácil implementar um sistema com qualidade.

Concluindo, como existe um vasto leque de ferramentas deste género que nos permitem realizar operações complexas de integração de dados, já não se justifica a criação manual de programas para executar estas operações.

3.2.1 Comparação entre Ferramentas de ETL *Open Source*

Existem muitas ferramentas de ETL, mas a maior parte são caras e assim optou-se por utilizar uma ferramenta *freeware* e *open source*. Atualmente existem várias ferramentas ETL *open source* como o CloverETL da Javlin [1], o Kettle da Pentaho [2] ou o Talend [3]. Os fornecedores destas ferramentas também têm versões comerciais do seu software, que utilizam o mesmo código *open source* e que oferecem mais funcionalidades como ferramentas de gestão e monitorização, e incluem suporte e garantia.

Estas três ferramentas têm um ambiente gráfico, de *drag and drop* e intuitivo. As três estão bem documentadas, têm um forte apoio da comunidade e têm imensas componentes já criadas que

permitem executar uma série de tarefas. Tanto o Talend como o CloverETL usam a framework do Eclipse [10] no seu editor do processo ETL. As três usam o JRE [24] para executar as transformações. As transformações do Talend também podem ser executadas como *scripts* Perl. O Talend é uma ferramenta que gera o código (Java ou Perl), compila-o e executa-o. O Kettle e o CloverETL são interpretadores de procedimentos ETL que são guardados em ficheiros XML.

É possível executar as transformações separadamente do editor com as três ferramentas. Como no Talend se pode compilar as transformações em pequenos pacotes Java ou *scripts* Perl, podemos desenvolvê-las e executá-las fora da plataforma. No Kettle e no CloverETL é necessário instalar algumas das suas bibliotecas.

O Talend usa Java, JavaScript ou Groovy como linguagem de script para fórmulas. O CloverETL usa CTL (Clover Transformation Language), que é uma linguagem de script especial para ETL e também pode usar Java. Já o Kettle usa Java ou Javascript. A empresa Talend e a Javlin estão mais focadas em soluções de integração de dados e gestão de dados, enquanto a Pentaho está focada em BI.

Para este projeto, qualquer uma das três soluções poderia ser usada. Por recomendação de um colaborador da I2S com experiência na utilização do Talend, optou-se por esta ferramenta.

Capítulo 4

Análise do Projeto

4.1 Análise do Departamento BI da i2S

A empresa i2S tem como clientes muitas companhias de seguros maioritariamente de Portugal e Angola e armazena muitas transações do negócio de cada uma delas nos seus sistemas OLTP. Assim, é importante que os seus clientes consigam consultar e analisar esses dados com bom desempenho e com garantia da sua integridade e veracidade, podendo assim tomar boas e suportadas decisões de negócio. O departamento de BI da i2S é responsável por criar bases de dados para consulta que armazenem informação de negócio a partir dos dados extraídos dos sistemas OLTP e de criar reporting analítico sobre essas bases de dados. Nesta secção, para contextualizar melhor o interesse deste projeto, parece-nos relevante caracterizar a atual solução de data warehousing disponibilizada pela i2S e a que poderá advir deste projeto, de acordo com o modelo de maturidade proposto pelo TDWI (The Data Warehousing Institute). O TDWI é um instituto educacional de business intelligence e data warehousing, e o modelo de maturidade é descrito por Wayne Eckerson, diretor de investigação no TDWI, em [8].

O modelo de maturidade mostra seis etapas que muitas das empresas seguem quando fazem evoluir as suas infraestruturas BI. Essas seis etapas são: pré-natal, infantil, criança, adolescente, adulto e sábio, como podemos ver na Figura 4.1. Neste modelo, à medida que se avança nas etapas o valor de negócio aumenta. As empresas evoluem a taxas diferentes ao longo destas seis etapas e cada uma delas pode exibir características de múltiplas etapas num dado período.

São seis as etapas, descritas de forma sucinta a seguir [8]:

- Etapa 1 (pré-natal) - as empresas apenas têm sistemas de gestão de *reporting* que geram um conjunto de relatórios estáticos;
- Etapa 2 (infantil) - as empresas usam *spreadmarts*, que são folhas de cálculo ou bases de dados *desktop*, para analisar os dados e que funcionam como substitutos dos data marts.

Cada spreadmart contém um conjunto de dados, métricas e regras únicas que não estão alinhadas com outros spreadmarts ou sistemas analíticos;

- Etapa 3 (criança) - as empresas já têm um pequeno conjunto de data marts independentes e utilizam ferramentas de *reporting* interativas (p.e. OLAP ou ferramentas de consulta *ad hoc*);
- Etapa 4 (adolescente) - existe uma padronização dos data marts que resulta num data warehouse que permite consultas ao longo de várias áreas de negócio. Nesta etapa as empresas utilizam aplicações de *dashboarding*;
- Etapa 5 (adulto) - as empresas já têm vários data warehouses distribuídos por vários departamentos ou áreas na empresa. Como estes DWs normalmente têm muitos dados inconsistentes, é nesta fase que as empresas decidem criar um data warehouse empresarial de modo a obterem uma "única versão da verdade". Nesta etapa, as empresas também desenvolvem scorecards em cascata que asseguram que toda a gente na empresa perceba qual é a estratégia e o seu papel na sua realização;
- Etapa 6 (sábio) - a empresa centra-se na melhoria contínua dos processos e na redução dos defeitos do sistema.

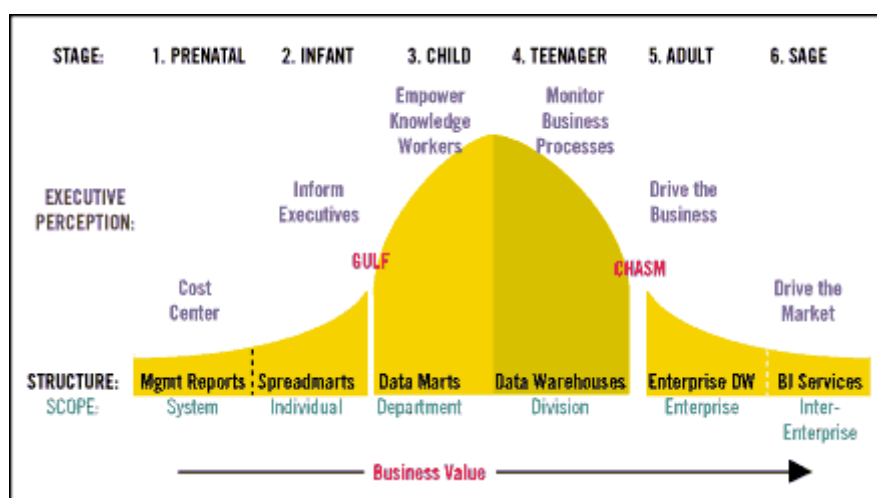


Figura 4.1: Modelo de Maturidade de Data Warehousing [8]

Atualmente, a maior parte das companhias de seguros clientes da i2S têm apenas uma ferramenta, desenvolvida pela i2S, chamada OCS que permite a extração de relatórios estáticos em Excel ou Crystal Reports, apresentando-se assim na etapa 1. Estes relatórios são manualmente programados sobre um ODS (Operational Data Store) e, por isso, existe uma maior propensão ao erro humano e a equipa de BI da i2S tem dificuldades em responder atempadamente aos pedidos dos clientes. Com este sistema, existe um enorme consumo de recursos (humanos e máquinas).

Na i2S, além do OCS, existe uma outra ferramenta, chamada qSIG, que pelas suas características já pode ser classificada como da etapa 3. Atualmente, apenas adquirida por três clientes, o qSIG é uma ferramenta de consulta *ad hoc* que permite aos utilizadores escolherem o que querem consultar e analisar na base de dados. A criação destas consultas é muito simples: inicialmente a ferramenta começa por apresentar uma lista de métricas para seleção, a seguir aparece uma lista das dimensões comuns às métricas anteriormente escolhidas também para seleção e no fim pode-se filtrar os valores dessas dimensões. Depois de executada a consulta, os resultados são retornados e exportados para uma *pivot table* no Microsoft Office Excel que é uma aplicação de folhas de cálculo.

O qSIG está sobre uma BD que armazena informação de negócio extraída de vários sistemas OLTP. A BD contém dados de vários processos de negócios mas é um data warehouse com problemas de desempenho nas consultas e as suas tabelas ocupam muito espaço em disco. Esta BD vai ser descrita com mais detalhe na próxima secção.

Surge assim o projeto em curso, descrito neste relatório, que vai permitir às seguradoras evoluírem para a quarta etapa. Os clientes vão poder ter um data warehouse para o setor de negócio de seguros não vida, que vai permitir análises ao longo de vários processos de negócio. Além disso, também vão poder aceder a este data warehouse a partir de uma das melhores plataformas de BI, o MicroStrategy, que oferece uma grande variedade de capacidades de BI, como vimos na Subsecção 2.1.2. Como resultado, com este projeto, as companhias de seguros vão exibir mais características da etapa 4 e vão ter um sistema de BI mais eficiente.

Para os clientes atingirem a etapa 5, a i2S terá que continuar a evoluir este data warehouse acrescentado mais informação de outros processos de negócio até obter um DW empresarial que satisfaça a maior parte das necessidades informacionais ou analíticas das seguradoras.

4.2 Análise dos Sistemas Operacionais e do Informacional

Na i2S existem três grandes setores de negócio que são responsáveis por desenvolver e dar suporte às atividades específicas nomeadamente da área administrativa financeira (ADF), da área de companhias de seguros não vida (GIS Não Vida) e da área de companhias de seguros vida (GIS Vida). Estes três setores de negócio precisam de um sistema de Business Intelligence que permita às companhias de seguros clientes consultar os seus dados e fazer análises de negócio eficientemente. O objetivo deste projeto é criar um sistema BI para o setor de negócio GIS Não Vida.

Atualmente na i2S, sobre estes sistemas operacionais, existe um processo de ETL em tempo real que extrai dados de todos os setores de negócios e junta-os numa única BD chamada Informacional. O Informacional é a BD consultada pela ferramenta qSIG e o sistema de base de dados utilizado é o DB2 UDB for iSeries [14].

Inicialmente o objetivo deste projeto era configurar a plataforma MicroStrategy para aceder diretamente ao Informacional.

Para o setor de negócio GIS Não Vida, o Informacional tem oito tabelas relacionais não normalizadas e cada uma delas guarda toda a informação relativa a um determinado processo de negócio:

1. Quatro tabelas que contêm as transações ocorridas em cada processo de negócio:
 - Tabela de Apólices (R#MOPAPO);
 - Tabela de Prémios/Estornos (Recibos) (R#MOPREC);
 - Tabela de Sinistros (R#MOPSIN);
 - Tabela de Indemnizações/Reembolsos (R#MOPIND);
2. Quatro tabelas de inventário, que registam o stock ou os eventos a decorrer com periodicidade mensal:
 - Tabela de Apólices em vigor (R#MOPAPOI);
 - Tabela de Prémios/Estornos (Recibos) em processo de cobrança/pagamento (R#MOPRECI);
 - Tabela de Sinistros abertos ou pendentes (R#MOPSINI);
 - Tabela de Indemnizações/Reembolsos em processo de pagamento/cobrança (R#MOPINDI);

O problema é que esta BD não cumpre dois requisitos fundamentais para o sucesso de um data warehouse [20]: simplicidade na organização dos dados de forma a que os utilizadores finais entendam o modelo facilmente; e um bom desempenho nas consultas quando os utilizadores executam os relatórios de negócio. O não cumprimento deste último requisito é evidenciado pelo qSIG, já que os utilizadores precisam de esperar muito tempo para ver os resultados das consultas. Logo, devido a estes problemas foi necessário optar pelo desenvolvimento de um data warehouse para este projeto.

4.3 Arquitetura do Data Warehouse

Escolher e definir a arquitetura do DW é um dos fatores chave de sucesso dos projetos de *data warehousing*. Uma má arquitetura pode levar a problemas como falta de escalabilidade, de desempenho e de veracidade dos dados [4]. No entanto, apesar do crescimento e do aumento de experiência no processo de data warehousing, ainda existem muitos desentendimentos em relação à melhor arquitetura. Existem quatro arquiteturas de data warehouse que lidam de forma diferente com o processo de ETL e na forma como os dados são armazenados no DW [4].

Uma das arquiteturas é a defendida por Bill Inmon chamada de arquitetura *hub and spoke* (arquitetura de data warehouse centralizada com data marts dependentes) ou arquitetura de data warehouse empresarial (EDW). Se os data marts dependentes forem lógicos (e não físicos) então chama-se somente arquitetura de data warehouse centralizada e como são similares podemos tratar estas duas arquiteturas como uma única [4]. A segunda é a arquitetura de data warehouse em *bus* ou a arquitetura de data marts ligados por dimensões conformes, defendida por Ralph Kimball. Estas arquiteturas de Inmon (Subsecção 4.3.3) e Kimball (Subsecção 4.3.2) são as mais relevantes. As outras duas, descritas nas Subsecções 4.3.1 e 4.3.4, são a arquitetura de data marts independentes e a arquitetura federada, e também são muito utilizadas por várias empresas e muito defendidas por várias pessoas.

4.3.1 Arquitetura de Data Marts Independentes



Figura 4.2: Arquitetura de data marts independentes

Na arquitetura ilustrada na Figura 4.2, os data marts são independentes de outros repositórios de dados e são criados para satisfazerem as necessidades locais de um departamento na empresa ou de um grupo de utilizadores e, por isso, não fornecem uma "única versão da verdade" de toda a empresa. Com esta arquitetura, é muito difícil fazer análises ao longo dos data marts porque eles costumam ter inconsistências nas definições dos dados e usam diferentes dimensões e factos (não conformes) [4]. O desenvolvimento desta arquitetura é simples, porém não deixa as empresas saírem da etapa 3 no modelo de maturidade do TDWI, descrito na Secção 4.1, porque não têm um DW integrado.

4.3.2 Arquitetura Data Warehouse em *Bus* (DWB)



Figura 4.3: Arquitetura de data marts ligados por dimensões conformes

A metodologia de desenvolvimento para a arquitetura de data warehouse em *bus*, representada na Figura 4.3, é a *bottom-up*. O desenvolvimento desta arquitetura começa pela definição inicial da "Matriz do DW em *Bus*" ou matriz de barramento do DW, com o objetivo de se fazer uma

análise sumária global de todos os processos e entidades de negócio importantes na empresa, resultando na identificação de todos os data marts e num barramento de dimensão conformes, representado nessa matriz. Nesta fase inicial todas as dimensões devem ser identificadas e o seu significado deve ser definido, para garantir uma evolução tranquila do projeto.

Após esta análise inicial escolhe-se um processo de negócio. Para este processo de negócio é criado um primeiro data mart, utilizando dimensões e factos conformes. Os próximos data marts devem utilizar estas dimensões conformes, de forma a que fiquem integrados, resultando no final num DW que permite uma visão empresarial dos dados. O modelo dimensional (esquema em estrela) é o modelo utilizado na conceção dos data marts e existem dados detalhados e sumarizados [4, 20]. O facto de existirem dados detalhados nos data marts traz a vantagem dos utilizadores não precisarem de transitar de um data mart para outra estrutura para obterem esses dados detalhados [9]. O objetivo do modelo esquema em estrela é otimizar a usabilidade e o desempenho das consultas, sendo esta uma das principais vantagens [20]. Como já vimos, outra das vantagens é que fornece informação de negócio rapidamente porque não é preciso estabelecer primeiro uma infraestrutura complexa [9].

Nesta arquitetura os data marts podem estar logicamente armazenados numa única BD física ou distribuídos pela empresa. A primeira opção minimiza a existência de dados redundantes e torna mais fácil a integração dos data marts, porque é mais fácil garantir dimensões conformes [9]. A segunda opção tem a desvantagem de poder haver uma tendência para a criação de data marts independentes, porque por vezes os departamentos ou unidades de negócios não cumprem as referências e regras para tornar as dimensões e factos conformes.

4.3.3 Arquitetura *Hub and Spoke*



Figura 4.4: Arquitetura de data warehouse empresarial (EDW)

A metodologia de desenvolvimento para a arquitetura hub and spoke, ilustrada na Figura 4.4, é a *top-down*. Nesta arquitetura, o DW contém dados no nível de detalhe mais alto e são mantidos na terceira forma normal. Data marts dependentes são depois criados tendo como origem dos dados o próprio DW [4]. Os DM dependentes são desenvolvidos para os departamentos ou áreas funcionais da empresa e podem ter estruturas de dados normalizadas, desnormalizadas ou multidimensionais.

Nesta arquitetura, por vezes, é utilizada uma área de estágio onde são guardados os dados recolhidos dos sistemas operacionais antes de serem carregados e integrados no DW [9].

A maior vantagem desta abordagem é que temos uma arquitetura integrada e flexível. Vamos obter um conjunto de data marts dependentes que estão consistentes e que permitem às empresas obterem uma "única versão da verdade". Os dados com grande detalhe presentes no DW também vão ser de grande utilidade porque permitem às empresas fazerem inúmeras análises e, possivelmente, encontrar novas e inesperadas necessidades de negócio. No entanto, isto também pode trazer uma desvantagem, já que pode não ser tão intuitivo para os utilizadores transitar de um data mart para o data warehouse quando estão à procura de detalhes sobre os dados sumarizados nos relatórios [9]. Outra desvantagem, já referida na Secção 2.2, é o facto desta abordagem poder ser mais dispendiosa e durar mais tempo a desenvolver que outras abordagens, especialmente no início.

4.3.4 Arquitetura Federada

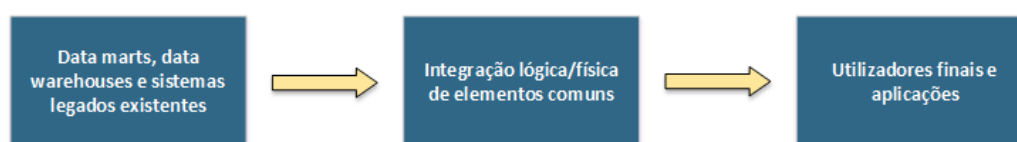


Figura 4.5: Arquitetura federada

Na arquitetura federada, representada na Figura 4.5, não existe um esforço por integrar o ambiente complexo de apoio à decisão existente numa única solução integrada. Esta arquitetura é mais uma solução realista para empresas que já tenham ambientes de apoio à decisão complexos e não querem ter que os reconstruir [4]. Esta abordagem, definida por Doug Hackney, apenas usa todos os meios para integrar os recursos de várias fontes de dados para atender às necessidades ou condições de negócio. Nesta arquitetura, as estruturas de apoio à decisão existentes permanecem então da mesma maneira e os dados são acedidos através de ferramentas que dão aos utilizadores uma visão global das diversas fontes de dados distribuídas, incluindo data warehouses, data marts e sistemas operacionais [27]. Essas ferramentas e métodos permitem consultas automáticas a essas fontes de dados, integram os resultados através de operações de junção, e apresentam-os aos utilizadores. Uma desvantagem desta abordagem é que pode perpetuar a descentralização e a fragmentação contínua dos recursos analíticos, tornando mais difícil a entrega de uma visão empresarial e podendo mesmo haver problemas de qualidade de dados e de desempenho [9]. Empresas com esta arquitetura estão normalmente na etapa 4 e 5 no modelo de maturidade do TDWI, descrito na Secção 4.1.

4.3.5 Abordagem Híbrida

As arquiteturas e as metodologias de desenvolvimento têm evoluído ao longo do tempo e têm ficado mais similares. Um exemplo disso, é a abordagem híbrida muito defendida por Pieter Mimno, que tenta retirar o melhor da abordagem *top-down* - a integração forçada por um data warehouse - e da abordagem *bottom-up* - rapidez e orientação do utilizador.

Nesta abordagem, começa-se por passar cerca de duas semanas a desenvolver um modelo empresarial na terceira forma normal antes de se desenvolver o primeiro data mart [9]. Estes data marts vão ser desenvolvidos utilizando modelos de esquema em estrela físicos. Esta abordagem utiliza ferramentas de ETL para armazenar e gerir os dados nos data marts e para sincronizar as diferenças existentes entre eles. Isto permite aos departamentos ou unidades de negócio nas empresas criarem as suas definições e regras para os dados que derivam do modelo empresarial sem sacrificar a integração a longo-termo. Estas ferramentas também são utilizadas para extrair os dados dos sistemas de origem e carregá-los no nível de detalhe e no nível sumarizado nos data marts [9].

O DW é desenvolvido depois de serem criados os primeiros data marts "dependentes". Isto acontece quando os utilizadores começam a pedir relatórios que contenham dados de detalhe de vários data marts. Este DW é então construído sobre os data marts e todos os dados detalhados são transferidos desses DMs para o DW. Os dados redundantes são então consolidados e assim poupa-se tempo, dinheiro e recursos de processamento [9].

4.4 Arquitetura Proposta para Este Projeto

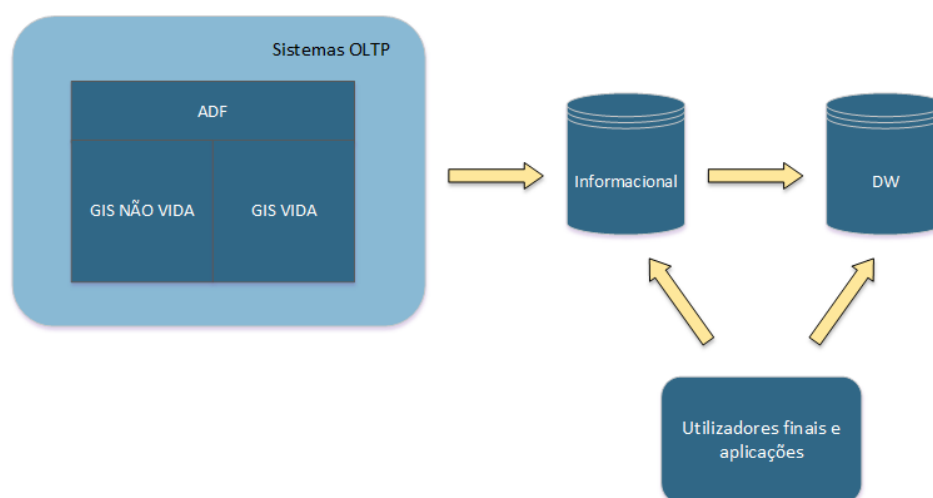


Figura 4.6: Arquitetura na i2S

Na Figura 4.6 podemos ver a arquitetura que foi definida para este projeto. A arquitetura mais parecida com a deste projeto é a defendida por Ralph Kimball e, por isso, a abordagem adotada foi a *bottom-up*. Foram desenvolvidos vários data marts dimensionais lógicos ligados por dimensões conformes numa única BD física e o sistema de base de dados é o SQL Server 2008. Porém, a arquitetura neste projeto não é totalmente similar à do DWB devido a alguns fatores.

A primeira diferença é a existência de outra base de dados de consulta no meio do processo de data warehousing. Como o Informacional já tem muita informação extraída dos sistemas operacionais, ficou decidido que o processo de ETL que iria desenvolver iria atuar sobre esta BD, em vez de ir diretamente aos sistemas operacionais. O Informacional já tem os dados todos necessários e desta forma poupa-se tempo de desenvolvimento. Outra das vantagens é que não existem grandes transformações a fazer. Os dados já foram todos limpos e transformados antes de serem carregados no Informacional. Contudo, não faz sentido ter esta BD intermédia que é consultada por outra ferramenta e no futuro o objetivo é implementar um processo que aceda diretamente aos sistemas operacionais.

Outro dos fatores foi a não construção de uma área de estágio, por solicitação da empresa. Dessa forma evitou-se trazer mais complexidade para o processo. Os dados teriam que passar pelo Informacional, pela a área de estágio e só depois é que chegavam ao data warehouse. A utilização do Informacional como área de estágio também não seria uma boa solução pois tornaria a BD confusa e como existem aplicações que acedem a essa BD, isso não é aconselhado. Mesmo não havendo grandes transformações para fazer, uma área de estágio trazia vantagens. Como as tabelas do Informacional não foram concebidas para este projeto, não estão preparadas para facilitar o carregamento dos dados no DW. Assim, o carregamento dos dados no DW não é direto, sendo necessárias junções entre tabelas. Sem uma área de estágio estas junções e as poucas transformações necessárias têm que ser feitas em memória.

Concluindo, apesar de ter adotado a abordagem *bottom-up*, o facto de se ter tido de utilizar o Informacional que já contém informação de negócio recolhida dos sistemas operacionais, não permitiu criar uma arquitetura DWB.

4.5 Levantamento de Requisitos

No que diz respeito ao levantamento de requisitos, houve uma série de reuniões com um cliente interessado neste projeto, para se perceber melhor as suas necessidades. Após algumas sessões, ficou decidido implementar, na primeira versão, sete relatórios base:

1. Estatística de apólices
2. Vendas totais por número de apólices

3. Apólices em vigor - evolução
4. Receita (prémios brutos emitidos, cobrados e em cobrança)
5. Comissões
6. Contagem de sinistros
7. Custos com sinistros

Na Secção 8 apresentam-se e explicam-se os relatórios de estatística de apólices, receita e de custos com sinistros. Os restantes relatórios encontram-se no Apêndice C.

Além destes relatórios também foram definidas várias métricas que estão no Apêndice C. Algumas dessas métricas são utilizadas nestes relatórios e as restantes apenas foram implementadas para permitir que os utilizadores finais possam desenvolver outros relatórios. Essas métricas podem ser agrupadas em cinco áreas: apólices, comissões, impostos, produção e sinistros.

Capítulo 5

Modelação Dimensional

O processo de conceção de um DM é composto por quatro etapas: seleção de um processo de negócio, definição da granularidade do processo de negócio, identificação das dimensões e identificação dos factos [20]. Este projeto envolve vários processos de negócio e, por isso, nesta secção, vou explicar o processo de conceção de vários data marts.

5.1 Processos de Negócio

Neste projeto foi implementado um DW para o setor de negócio GIS Não Vida e existem quatro processos de negócio essenciais neste setor:

- apólices,
- recibos (prémios/estornos),
- sinistros e
- indemnizações/reembolsos.

Para a ser possível implementar as métricas e relatórios requisitados, foram criados quatro esquemas em estrela, para cada um destes processos de negócio, e integrados um-a-um no DW.

Estes processos de negócio são todos essenciais no DW porque qualquer companhia de seguros está interessada em fazer análises que envolvam as transações das apólices e os respetivos prémios/estornos, e as transações dos sinistros e as respetivas indemnizações/reembolsos. Os utilizadores querem relatórios e análises ad-hoc que cruzem as métricas resultantes destes quatro processos de negócio. Por exemplo, querem medir o lucro gerado ao longo do tempo, por ramo, por cobertura, por distrito, etc. O cálculo do lucro da companhia de seguros só é

possível se subtrairmos os custos à receita. Logo, o DW tem que conter estes quatro processos de negócios, antes de se implementar as métricas e relatórios.

Neste DW também vamos ter quatro processos de negócio de inventário de forma a se conseguir medir os níveis de inventário todos os meses:

- apólices de inventário,
- recibos (prémios/estornos) de inventário,
- sinistros de inventário e
- indemnizações/reembolsos de inventário .

Por exemplo, os utilizadores vão querer medir o prémio das apólices em vigor, quantos sinistros estão em curso ou o valor das indemnizações em pagamento para um determinado período. As tabelas de factos destes processos de negócio são de sumarização periódica.

No DW, uma tabela de factos de inventário vai ter as mesmas dimensões e muitos dos mesmos factos presentes na tabela de factos transaccional, sendo mesmo a única diferença o tipo de tabela. No entanto, no Informacional, estas tabelas não são abastecidas a partir das tabelas que guardam as transações, mas sim a partir de tabelas diferentes existentes nos sistemas operacionais, podendo mesmo existir apólices ou sinistros nestas tabelas de inventário que não existem nas tabelas transacionais, apesar de estranho e pouco provável.

5.2 Definição da Granularidade

Nesta segunda etapa devemos escolher a granularidade, ou seja, devemos definir que nível de detalhe dos dados devemos disponibilizar em cada DM. Quanto mais detalhe houver, mais baixo será o nível de granularidade. Para definir a granularidade temos que definir claramente o que vai significar um registo de uma tabela de factos [20].

- Tabela de factos FACTO_APOLICE: um registo para cada transação de uma cobertura de uma apólice.
- Tabela de factos FACTO_RECIBO: um registo para cada transação de um prémio ou estorno associado a uma cobertura de uma apólice.
- Tabela de factos FACTO_SINISTRO: um registo para cada transação de um sinistro associado a uma cobertura de uma apólice.
- Tabela de factos FACTO_IND_REEMB: um registo para cada transação de uma indemnização ou reembolso associada(o) a uma cobertura de uma apólice.

- Tabela de factos FACTO_APOLICE_INV: um registo para cada cobertura numa apólice em vigor em cada mês.
- Tabela de factos FACTO_RECIBO_INV: um registo para cada prémio/estorno em cobrança/pagamento associado a uma cobertura de uma apólice em cada mês.
- Tabela de factos FACTO_SINISTRO_INV: um registo para cada sinistro aberto/pendente associado a uma cobertura de uma apólice em cada mês.
- Tabela de factos FACTO_IND_REEMB_INV: um registo para cada indemnização/reembolso em pagamento/cobrança associada(o) a uma cobertura de uma apólice em cada mês.

A granularidade nestas tabelas de factos é a mais baixa possível. Uma maneira de aumentar a granularidade, por exemplo da tabela de FACTO_APOLICE, seria não ter os registos por cobertura. Assim, por exemplo, para uma apólice com cinco coberturas, em vez de termos cinco registos na tabela para esta apólice, passaríamos a ter só um registo e os factos seriam todos agregados.

Preferencialmente o nível de detalhe deve ser o mais alto possível, de modo a que não haja mais formas de subdividir os dados [20]. Para termos um sistema de BI eficiente, os utilizadores devem poder navegar nos dados até à informação mais detalhada possível (mais flexibilidade). Além disso, quanto mais detalhados forem os factos, mais dimensões conseguem utilizar nas análises (mais dimensionalidade). Só assim vão poder analisar os dados de todas as formas possíveis e retirar o máximo de informação acerca dos dados [20].

Por outro lado, como têm menos registos, as tabelas com granularidade alta ocupam menos espaço em disco e existe um melhor desempenho nas consultas no DW. Quando o objetivo do DW for permitir a realização de análises que não necessitam de um baixo nível de granularidade, então o armazenamento de informação com alto nível de detalhe poderá não fazer sentido.

Porém, neste projeto, não se quis limitar as análises dos utilizadores para que eles consigam fazer o *drill down* para níveis mais altos de detalhe dos dados. Por isso, optou-se pela mais baixa granularidade possível para garantir sempre o máximo de dimensionalidade e flexibilidade. É natural que a maior parte dos utilizadores não pretenda analisar em detalhe as medidas para cada cobertura de cada apólice. No entanto, podem querer ver, por exemplo, na tabela de FACTO_SINISTRO, em que coberturas ocorreram mais sinistros no ano de 2012. Isto não seria possível se tivéssemos informação resumida, onde cada linha na tabela estivesse ao nível da apólice associada ao sinistro.

5.3 Identificação das Dimensões

As dimensões identificadas para este projeto estão listadas na Tabela 5.1.

Dimensões	
DATA	APÓLICE EVENTO
REGIÃO	RECIBO EVENTO
COBERTURA	SINISTRO EVENTO
TITULAR	INDEMNIZAÇÃO EVENTO
ESTRUTURA COMERCIAL	APÓLICE ESTADO
APÓLICE	RECIBO ESTADO
RECIBO	SINISTRO ESTADO
SINISTRO	INDEMNIZAÇÃO ESTADO
INDEMNIZAÇÃO	

Tabela 5.1: Dimensões incluídas nos DMs

Data

Uma das primeiras dimensões escolhidas foi a DATA, necessária em qualquer DW, pois permite contextualizar os factos no tempo, possibilitando analisar o processo numa perspetiva temporal. A dimensão DATA apenas contém a data do calendário e não o evento no dia, pois neste negócio não traz grandes benefícios analisar a informação por hora/minuto/segundo. Possui um registo por cada dia do calendário civil e duas hierarquias, como se pode verificar na Figura 5.1.

Região

Representada na Figura 5.2, a dimensão REGIÃO possui uma única hierarquia com três níveis, permitindo analisar os valores das métricas por concelho ou distrito de um país. O código postal é o último nível e é onde é guardada informação sobre o local onde uma apólice ou sinistro foi registado.

Cobertura

A dimensão COBERTURA é necessária em todos os processos de negócio. A dimensão COBERTURA descreve todas as coberturas das apólices existentes. Como é possível visualizar na Figura 5.3, esta dimensão possui uma hierarquia e dois níveis onde se armazena a informação sobre a cobertura e ramo contabilístico.

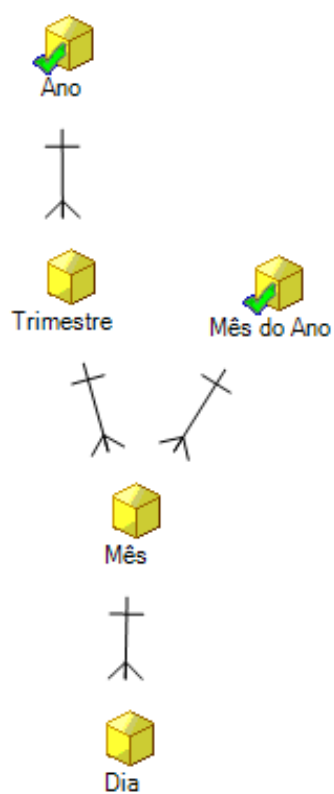


Figura 5.1: Hierarquias e níveis da dimensão Data



Figura 5.2: Hierarquia e níveis da dimensão Região



Figura 5.3: Hierarquia e níveis da dimensão Cobertura

Titular

A dimensão TITULAR armazena informação sobre os titulares das apólices. Esta dimensão é necessária para ser possível compreender e analisar o comportamento de certos segmentos de titulares. As hierarquias e níveis desta dimensão estão ilustrados na Figura 5.4.

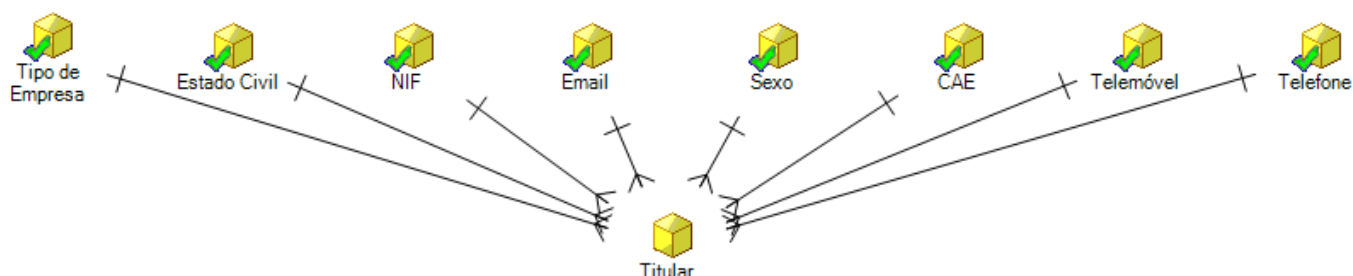


Figura 5.4: Hierarquias e níveis da dimensão Titular

Estrutura Comercial

A dimensão ESTRUTURA COMERCIAL indica quem foram os intervenientes que produziram a apólice ou sinistro desde o agente produtor (nível mais fino possível de granularidade) até à companhia de seguros (nível mais grosso). Esta dimensão tem uma hierarquia e o número de níveis varia conforme a companhia de seguros cliente da i2S. Na Figura 5.5 podemos verificar a estrutura comercial de um dos clientes da i2S. Nesta companhia de seguros a dimensão possui 7 níveis.

Apólice, Recibo, Sinistro e Indemnização

A dimensão APÓLICE está presente em todos os processos de negócio e contém informação sobre as apólices. O nível mais fino possível desta dimensão é a própria apólice que é identificada pelo seu código. Esta dimensão possui três hierarquias: uma para as apólices temporárias, outra para as apólices anuais e a terceira relativa ao tipo de apólice, que para além do código da apólice também é composta pelos níveis Ramo e Produto, como podemos verificar na Figura 5.6.

Outras dimensões identificadas foram a RECIBO, SINISTRO e INDEMNIZAÇÃO, que só estão mapeadas com algumas tabelas de factos. Existe a dimensão RECIBO que contém todos os atributos de recibo e que está mapeada com as tabelas de factos de recibo através do código do recibo. A dimensão SINISTRO que está mapeada com as tabelas de factos de sinistro e indemnização através do código do sinistro, pois uma indemnização está sempre associada a um sinistro. E a dimensão INDEMNIZAÇÃO que só está mapeada com a tabela de factos de indemnização através do código da indemnização.



Figura 5.5: Hierarquia e níveis da dimensão Estrutura Comercial

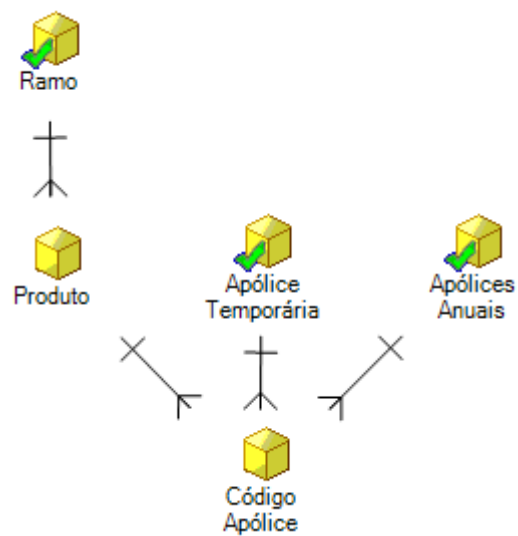


Figura 5.6: Hierarquias e níveis da dimensão Apólice

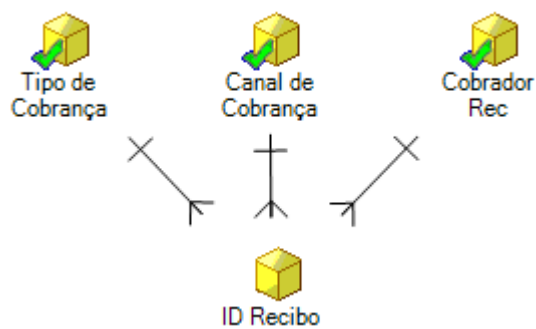


Figura 5.7: Hierarquias e níveis da dimensão Recibo Evento

Apólice Evento, Recibo Evento, Sinistro Evento e Indemnização Evento

Além das últimas quatro dimensões também são necessárias as dimensões APÓLICE EVENTO, RECIBO EVENTO, SINISTRO EVENTO e a INDEMNIZAÇÃO EVENTO. Estas dimensões servem para armazenar os atributos cujos valores podem variar conforme o tipo da transação/evento. Por exemplo, na tabela de factos FACTO_APOLICE, se o tipo da transação for uma anulação da apólice, estes atributos podem ter valores diferentes dos do evento de criação dessa mesma apólice. Assim, estes atributos não poderiam estar associados ao código da apólice/recibo/sinistro/indemnização mas sim ao identificador (ID) da apólice e do tipo de evento/transação. Na Figura 5.7 temos as 3 hierarquias e 4 níveis da dimensão RECIBO EVENTO. Todas as outras dimensões de evento são deste género e não existem hierarquias compostas por mais de 2 níveis.

Apólice Estado, Recibo Estado, Sinistro Estado e Indemnização Estado

Por fim, também existem as dimensões APÓLICE ESTADO, RECIBO ESTADO, SINISTRO ESTADO e a INDEMNIZAÇÃO ESTADO que são do tipo de transação ou do estado da(o) apólice/recibo/sinistro/indemnização. Estas dimensões possuem apenas um nível. A Figura 5.8 ilustra o nível da dimensão APÓLICE ESTADO.



Figura 5.8: Hierarquia e nível da dimensão Apólice Estado

Na Secção 5.4 apresentamos em detalhe o modelo, sendo possível perceber melhor que dimensões cada uma das tabelas de factos tem. Na Secção 6.4 vão ser identificados os atributos de cada um dos níveis destas dimensões.

5.4 Matriz do Data Warehouse em *Bus* e Dimensões Conformes

Definidas as dimensões e as tabelas de factos podemos recorrer à "Matriz do Data Warehouse em *Bus*", ilustrada na Figura 5.9, para percebermos melhor que dimensões estão associadas a cada processo de negócio e a interação existente entre os processos de negócio através de dimensões comuns.

As linhas da matriz na Figura 5.9 representam os data marts da empresa e cada um corresponde a um processo de negócio. As colunas da matriz representam todas as dimensões existentes.

Processos de negócio	Dimensões comuns																
	Data	Região	Estrutura comercial	Titular	Cobertura	Apólice	Apólice evento	Recibo	Recibo evento	Sinistro	Sinistro evento	Indemnização	Indemnização evento	Apólice estado	Recibo estado	Sinistro estado	Indemnização estado
	X	X	X	X	X	X	X							X			
	X	X	X	X	X	X	X							X			
	X	X	X	X	X	X		X	X						X		
	X	X	X	X	X	X		X	X		X	X				X	
	X	X	X	X	X	X											
	X	X	X	X	X	X											
	X	X	X	X	X	X											
	X	X	X	X	X	X											
	X	X	X	X	X	X											
	X	X	X	X	X	X											

Figura 5.9: Matriz do Data Warehouse em *Bus*

Depois de marcar as células indicando que existe uma relação entre a dimensão e o processo de negócio, conseguimos perceber que existem dimensões comuns entre processos de negócio.

Como neste projeto se tentou adotar a arquitetura de Ralph Kimball (arquitetura de data warehouse em *bus*), teve-se que garantir que todos os data marts se relacionassem entre si, para evitar que o resultado fosse um conjunto de data marts independentes. Para ser possível relacionar esses data marts, as dimensões comuns devem ser conformes e devem suportar a capacidade de integrar e analisar dados provenientes de múltiplos processos de negócio. Para serem consideradas conformes, deve-se assegurar nas tabelas de dimensão que as chaves, nomes das colunas, definições dos atributos e os valores dos atributos são consistentes em todos os processos de negócio [20].

Para ligar estes DMs e criar então um DW podemos usar esta matriz que serve como ferramenta tanto no planeamento como na comunicação. Ao analisar a matriz, podemos perceber quais as dimensões que precisam de maior atenção na questão da conformidade devido à sua participação em vários data marts. Em geral, esta matriz também pode ser eficaz na comunicação entre as diferentes equipas que implementam data marts porque é um resumo que transmite o plano de todo o DW.

Neste setor de negócio Não Vida, os data marts estão logicamente dispostos dentro de uma única base de dados física e, assim, a integração dos DM foi feita facilmente, já que uma dimensão conforme é uma única tabela física na base de dados. No entanto, na i2S, poderá haver no futuro múltiplas bases de dados para este ou até para outros setores de negócio. Neste caso, as dimensões vão estar duplicadas e devem estar sincronizadas em cada DM. Em ambos os casos, as dimensões conformes devem ser manipuladas uma única vez no processo de ETL, podendo depois ser replicadas ou reutilizadas por múltiplas tabelas de factos [11].

Por vezes, as tabelas de factos necessitam de estar associadas a dimensões com um nível de granularidade mais alto, por exemplo, as tabelas sumarizadas com factos agregados. Isto pode resultar na existência de duas dimensões iguais que estão definidas com níveis de granularidade distintas no mesmo ou em diferentes processos de negócio. Nestes casos, não podemos partilhar estas dimensões como tabelas únicas coincidentes ao longo dos processos de negócio porque a granularidade é diferente. No entanto, as tabelas de dimensão podem ser ainda conformes, sendo que uma é um subconjunto da outra. Os atributos partilhados pelas duas tabelas de dimensão, devem ter os mesmo nomes, definições e valores [20].

Neste projeto, só aconteceu um caso deste género. Os processos de negócio de inventário estão por mês, enquanto os restantes processos de negócio estão por dia. Neste caso, vai ter que haver um subconjunto da tabela de dimensão de DATA tanto nas linhas (alguns valores são filtrados) como nas colunas (alguns atributos não existem). Esse subconjunto de atributos pode ser chamado de dimensão MÊS. A tabela desta dimensão comparada com a da dimensão DATA, apenas não vai ter o atributo DIA, porque não existe mais nenhum atributo na tabela da dimensão DATA que não se aplica na granularidade mensal, como vamos ver nas Figuras

6.6 e 6.7 da Secção 6.4. Os restantes atributos, que vão ser partilhados pelas duas tabelas de dimensão, vão ter os mesmo nomes, definições e valores. Outra solução que também poderia ser aplicada neste caso, seria normalizar a tabela de dimensão DATA, em vez de criar uma tabela de dimensão MÊS. Esta abordagem vai ser explicada na Secção 6.6.

Neste projeto, os processos de negócio foram então todos integrados e as dimensões são todas conformes. No futuro, caso seja necessário implementar mais data marts tanto nesta base de dados como noutra qualquer, deve-se então incluir esses processos de negócio no DW, efetuando as alterações necessárias às dimensões existentes e acrescentando novas dimensões.

5.5 Identificação dos Factos

As tabelas de factos e os factos contidos são produto do levantamento de requisitos que permitiu concluir que neste DW é necessário existir pelo menos oito tabelas de factos, sendo quatro delas transacionais e outras quatro de sumarização periódica. Essas oito tabelas vão ter os seguintes factos, todos numéricos e aditivos:

Tabela de factos FACTO_APOLICE

- CAPITAL_APOLICE: valor do capital da apólice, que representa o valor máximo que a companhia de seguros paga em caso de sinistro.
- PREMIO_APOLICE: valor do prémio da apólice, que representa o valor total, incluindo taxas e impostos, que o tomador do seguro deve pagar à companhia de seguros pelo seguro.
- CARTA_VERDE_COM_E_SEM_SELO: custo referente à emissão deste documento, possivelmente acrescido do custo de envio deste documento por correio.
- JUROS_FRACCIONAMENTO: valor que acresce ao prémio caso o tomador do seguro opte por pagá-lo em prestações.
- CUSTO_APOLICE: custo referente à emissão deste documento.
- FGA: taxa parafiscal a incidir sobre os prémios comerciais, existente nos contratos do ramo automóvel. Taxa de 2,5% respeitante ao seguro obrigatório de responsabilidade civil automóvel. (Fundo de Garantia Automóvel)
- PREVENCAO_RODOVIARIA: taxa parafiscal a incidir sobre os prémios comerciais, existente nos contratos do ramo automóvel. Taxa de 0,21%. (Prevenção Rodoviária)
- INEM: taxa parafiscal a incidir sobre os prémios comerciais, existente nos contratos de acidentes de trabalho, acidentes Pessoais, doença e automóvel. Taxa de 1%. (Instituto Nacional de Emergência Médica)

- SNB_ANPC: taxa parafiscal a incidir sobre os prémios comerciais, existente nos contratos de incêndio, multiriscos, agrícola e mercadorias transportadas com taxas de valores de 3,9, 6 ou 13%. (SNBPC - Serviço Nacional de Bombeiros e Proteção Civil)
- FAT: taxa parafiscal a incidir sobre os salários seguros, existente nos contratos do ramo acidentes de trabalho. Taxa de 0,15%.
- SELO: imposto indireto existente em todos os ramos, com taxas de valores de 3, 5 ou 9%.
- NUMERO_DE_PESSOAS_SEGURAS: número de pessoas cuja vida, saúde ou integridade física se segura.

Os valores dos factos supra referenciados, são recolhidos dos sistemas operacionais e aí calculados conforme acima explicado.

Tabela de factos FACTO_APOLICE_INV

- PREMIO_APOLICE_APO_I: valor total do prémio da apólice.
- CARTA_VERDE_COM_E_SEM_SELO_APO_I: custo referente à emissão deste documento.
- JUROS_FRACCIONAMENTO_APO_I: valor que acresce ao prémio caso o tomador do seguro opte por pagá-lo em prestações.
- CUSTO_APOLICE_APO_I: custo referente à emissão deste documento.
- FAT_APO_I: valor referente à taxa FAT.
- FGA_APO_I: valor referente à taxa FGA.
- INEM_APO_I: valor referente à taxa INEM.
- PREVENCAO_RODOVIARIA_APO_I: valor referente à taxa de prevenção rodoviária.
- SNB_ANPC_APO_I: valor referente à taxa SNBPC.
- SELO_APO_I: valor referente a um imposto indireto existente em todos os ramos.
- NUMERO_DE_PESSOAS_SEGURAS_APO_I: quantidade de pessoas cuja vida, saúde ou integridade física se segura.

Tabela de factos FACTO_RECIBO

- TOTAL_PREMIO_REC: valor total do prémio/estorno constante no recibo. Se for prémio, representa o valor total, incluindo taxas e impostos, que o tomador do seguro deve pagar à companhia de seguros pelo seguro. Se for estorno, representa o valor que a companhia de seguros deve devolver ao tomador do seguro, de uma parte do prémio já pago, nomeadamente no caso do contrato de seguro terminar antes do seu fim.

- TOTAL_COMISSOES: remuneração do mediador de seguros pela atividade de mediação.
- COMISSOES_COSSEGURO: remuneração do mediador de seguros pela atividade de mediação nos contratos em regime de cosseguro.
- CARTA_VERDE_COM_E_SEM_SELO_REC: custo constante no recibo, referente à emissão deste documento, possivelmente acrescido do custo de envio deste documento por correio.
- CARTA_VERDE_SEM_SELO_REC: custo constante no recibo, referente à emissão deste documento.
- CUSTO_APOLICE_REC: valor deste custo de emissão constante no recibo.
- NOVO_FGA_REC: valor no recibo referente à taxa FGA.
- FAT_REC: valor desta taxa constante no recibo.
- INEM_REC: valor desta taxa constante no recibo.
- JUROS_FRACCIONAMENTO_REC: valor constante no recibo que acresce ao prémio caso o tomador do seguro opte por pagá-lo em prestações.
- JUROS_FRACCIONAMENTO_REC_COS: valor constante no recibo que acresce ao prémio do contrato em regime de cosseguro caso o tomador do seguro opte por pagá-lo em prestações.
- PBE_REC: valor do prémio comercial do recibo acrescido dos custos de emissão do contrato (apólice/acta adicional, carta verde, certificado de seguro e juros de fraccionamento). (Prémio bruto emitido)
- PREMIO_COMERCIAL_REC: valor do prémio comercial no recibo, que representa o custo das coberturas do contrato, acrescido de custos de aquisição, gestão e cobrança.
- PREMIO_COMERCIAL_REC_COS: valor do prémio comercial no recibo, que representa o custo das coberturas do contrato em regime de cosseguro, acrescido de custos de aquisição, gestão e cobrança.
- PREVENCAO_RODOVIARIA_REC: valor desta taxa constante no recibo.
- ANPC_REC: valor no recibo referente à taxa SNBPC.
- SELO_REC: valor deste imposto indireto constante no recibo.

Tabela de factos FACTO_RECIBO_INV

- ANPC_REC_I: valor no recibo referente à taxa SNBPC.
- NOVO_FGA_REC_I: valor no recibo referente à taxa FGA.
- FAT_REC_I: valor no recibo referente à taxa FAT.
- INEM_REC_I: valor no recibo referente à taxa INEM.

- **PBE_REC_I**: valor do prémio comercial do recibo acrescido dos custos de emissão do contrato. (Prémio bruto emitido)
- **PREVENCAO_RODOVIARIA_REC_I**: valor desta taxa constante no recibo.
- **SELO_REC_I**: valor deste imposto indireto constante no recibo.
- **TOTAL_VALOR_REC_I**: representa o valor total do prémio/estorno em cobrança/pagamento constante no recibo.

Tabela de factos FACTO_SINISTRO

- **VALOR_SINISTROS**: valor da provisão para sinistros, que representa o custo estimado para a regularização integral do sinistro.

Tabela de factos FACTO_SINISTRO_INV

- **CONSTITUICAO_PROVISAO_SINISTRO_I**: custo estimado para a regularização integral do sinistro.
- **AUMENTO_PROVISAO_SINISTRO_I**: variação positiva do custo estimado da provisão.
- **DIMINUICAO_PROVISAO_SINISTRO_I**: variação negativa do custo estimado da provisão.
- **PROVISAO_INDEMNIZACAO**: custo estimado das indemnizações a suportar.
- **PROVISAO_REEMBOLSO**: custo estimado das indemnizações a suportar.
- **INDEMNIZACAO_LIQUIDADA**: custo total das indemnizações pagas.
- **REEMBOLSO_LIQUIDADO**: proveito total dos reembolsos recebidos.

Tabela de factos FACTO_INDEMNIZACAO

- **VALOR_INDEMNIZACAO**: valor da indemnização/reembolso. Se for uma indemnização, representa o valor devido pela seguradora para pagar um dano resultante de uma situação coberta pela apólice. Se for um reembolso, representa o valor devido pelo responsável pelo dano à seguradora, referente a uma indemnização paga por essa seguradora ao beneficiário do contrato.

Tabela de factos FACTO_INDEMNIZACAO_INV

- **VALOR_INDEMNIZACAO_IND_I**: valor da indemnização/reembolso em pagamento/cobrança.

Capítulo 6

Desenvolvimento do Data Warehouse

6.1 Tabelas de Factos Transacionais

No DW existem quatro tabelas de factos transacionais e podemos ver os seus campos nas Figuras 6.1 - 6.4. Estas quatro figuras também ilustram quatros esquemas em estrela que constituem um subconjunto do modelo dimensional do DW. Todas estas tabelas de factos têm pelo menos dez chaves estrangeiras, designadas pela notação "FK", que se conectam às chaves primárias das tabelas de dimensão.

As chaves primárias nas tabelas de factos são geralmente constituídas pela totalidade, ou por um subconjunto, das chaves estrangeiras e, por isso, chamam-se chaves compostas. Neste projeto, em todas as tabelas de factos, a chave primária é apenas constituída por um subconjunto das chaves estrangeiras porque só é necessário esse subconjunto para tornar cada linha numa tabela de factos única. Por exemplo, no caso da tabela FACTO_APOLICE, esses campos são o ID_APOLICE e o KEY_COBERTURA. Outra solução na definição da chave primária, seria criar um novo campo nas tabelas de factos para servir como chave primária única. Contudo, isso não traria grandes vantagens porque essa chave não seria utilizada para criar relações. A tabela de factos apenas ficaria maior e tornar-se-ia inútil inserir um índice nessa chave primária.

Na tabela de factos transacional FACTO_APOLICE encontramos uma dimensão que se chama RISCO_CONTA_PARA_CAPITAL_APOLICE. Esta dimensão é considerada uma dimensão degenerada, designada pela notação "DD", porque não tem uma tabela de dimensão. Esta dimensão está no nível de granularidade mais baixo e está associada a uma cobertura de uma apólice. Nas coberturas de uma apólice nem todos os valores no facto CAPITAL_APOLICE associado contam quando se quer descobrir o capital dessa apólice. Existem valores do facto CAPITAL_APOLICE associados a certas coberturas que não devem ser somados. Nesses registos dessas coberturas o campo RISCO_CONTA_PARA_CAPITAL_APOLICE tem o valor "0". Nos casos onde a cobertura conta para o capital da apólice então esta dimensão tem o valor "1".

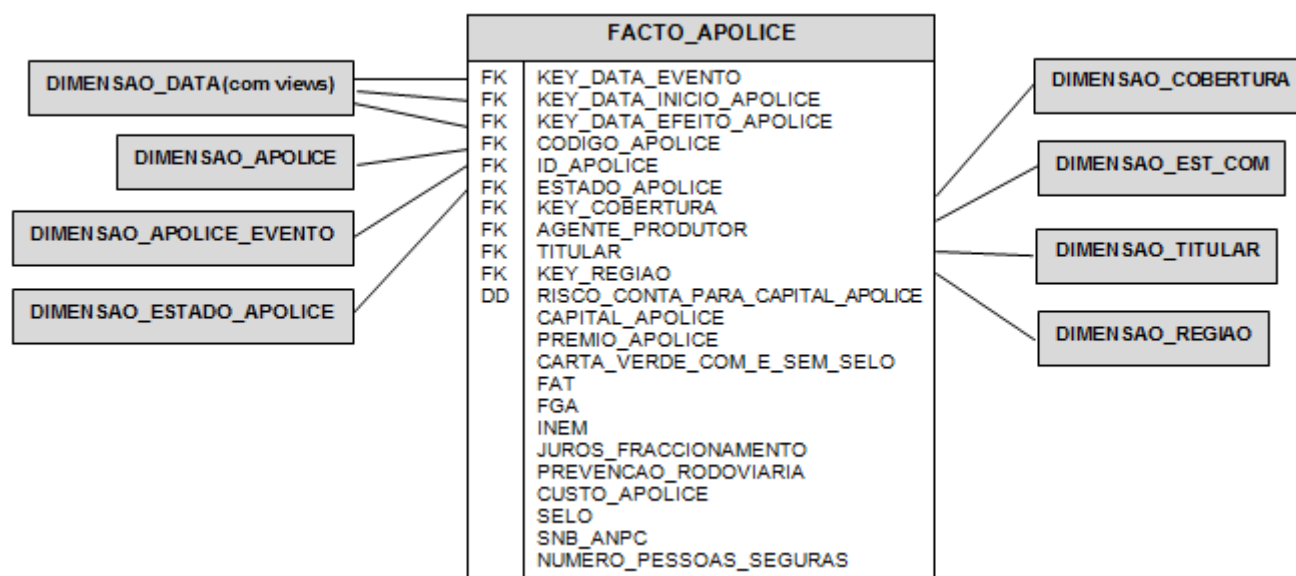


Figura 6.1: Tabela de Factos de Apólice

Normalmente, cada uma das datas é uma chave estrangeira na tabela de factos. Contudo, não devemos ligar duas ou mais chaves estrangeiras à mesma tabela de dimensão DATA. O sistema de base de dados iria interpretar estas datas como sendo idênticas, o que pode não ser verdade. Para resolver este problema, podemos usar *views*(SQL). Com as *views* já conseguimos obter dimensões independentes para as datas. Porém, em cada *view* os atributos devem ter nomes únicos. Por exemplo, na *view* da data de início da apólice (Código 6.1), o atributo DIA tem que ter um nome diferente do atributo DIA da *view* da data de evento. Se isto não acontecer, corremos o risco de não saber a que dimensão corresponde o atributo quando estamos a analisar um relatório.

Código 6.1: Código utilizado na criação da *view* da data de início da apólice

```
CREATE VIEW [dbo].[DIMENSAO_DATA_INICIO_APOLICE] (
KEY_DATA_INICIO_APOLICE, DIA_INICIO_APOLICE,
ANO_INICIO_APOLICE, MES_DO_ANO_INICIO_APOLICE,
MES_DO_ANO_INICIO_APOLICE_DESC, MES_INICIO_APOLICE,
MES_INICIO_APOLICE_DESC, TRIMESTRE_INICIO_APOLICE,
TRIMESTRE_INICIO_APOLICE_DESC)
AS
SELECT KEY_DATA, DIA, ANO, MES_DO_ANO, MES_DO_ANO_DESC,
MES, MES_DESC, TRIMESTRE, TRIMESTRE_DESC
FROM dbo.DIMENSAO_DATA
```

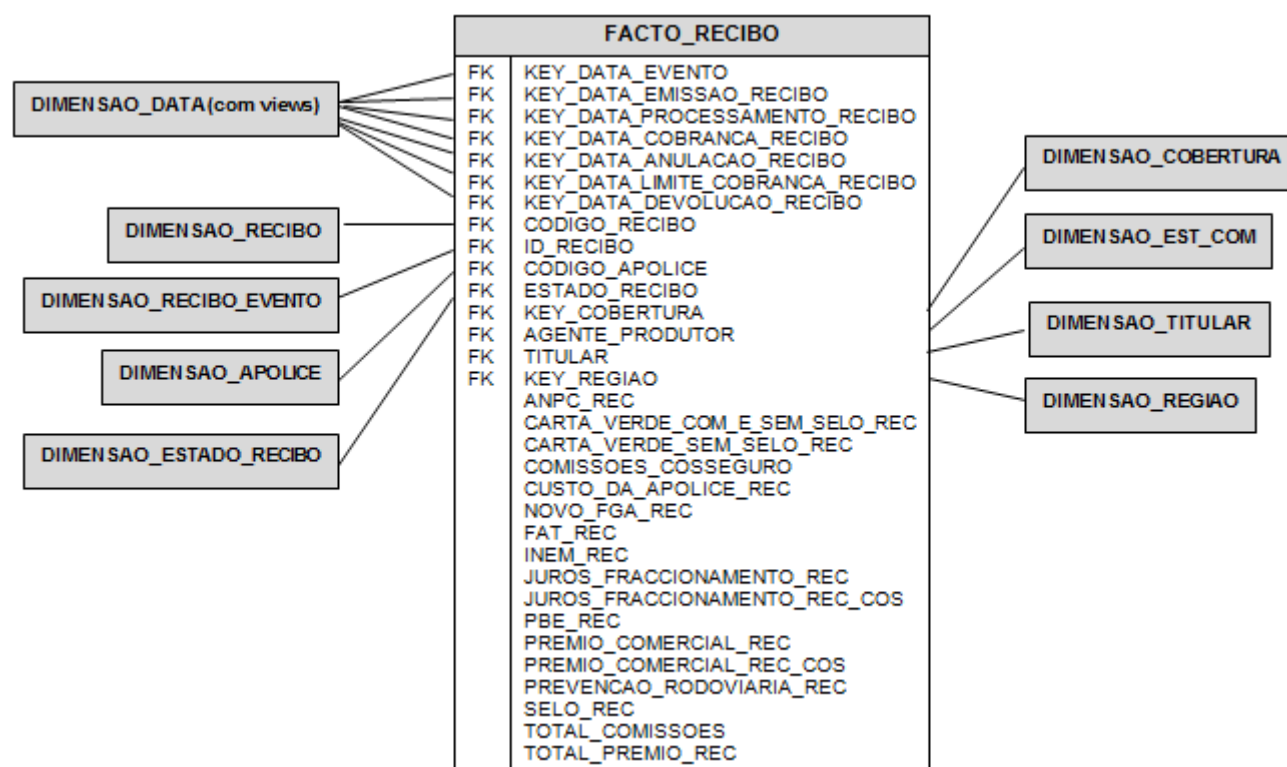



Figura 6.2: Tabela de Factos de Recibo

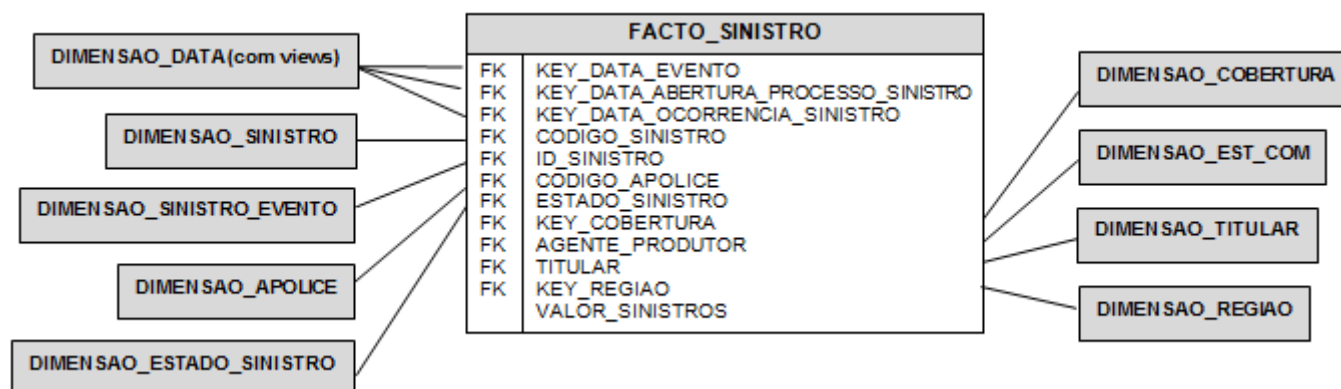


Figura 6.3: Tabela de Factos de Sinistro

Inicialmente tentou-se juntar os factos da tabela das indemnizações com as dos sinistros mas rapidamente nos apercebemos de que isso traria desvantagens mais significativas que vantagens. As vantagens seriam que o desenho da base de dados ficava mais simples e intuitivo e deixaria de haver uma junção de tabelas nas consultas que precisassem de factos de sinistro e indemnização. No entanto, o processo de ETL demoraria muito mais tempo, porque iríamos precisar de fazer a junção entre os dados de sinistros e os dados das indemnizações. Como as tabelas no Informacional não estão normalizadas, a junção entre estas duas tabelas que podem ter milhões registos, mesmo com índices, ia ter baixo desempenho. Outra grande desvantagem

seria o tamanho das tabelas de factos resultantes no DW, já que um sinistro pode ter múltiplas indemnizações que podem ser tantas quanto os lesados e/ou a natureza dos danos.

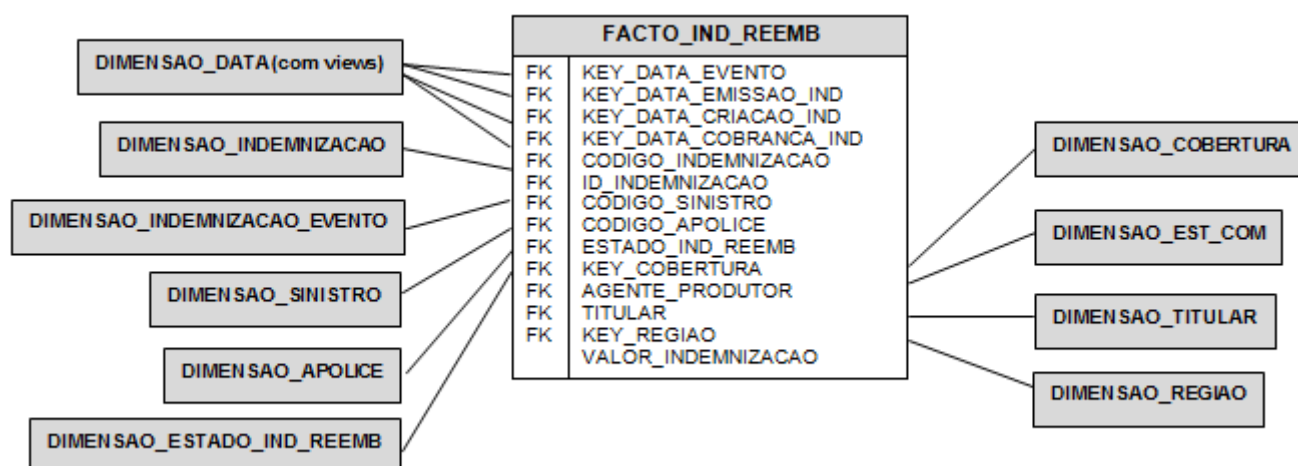


Figura 6.4: Tabela de Factos de Indemnização/Reembolso

6.2 Tabelas de Factos de Sumarização Periódica

As quatro tabelas de inventário correspondem a tabelas de factos de sumarização periódica, onde um registo se repete a cada período, enquanto o que está a ser medido existir. Por exemplo, na tabela FACTO_APOLICE_INV, um registo repete-se todos os meses enquanto a apólice estiver em vigor. Desta forma, quando se consulta esta tabela, consegue-se descobrir, por exemplo, qual é o prémio das apólices em vigor num determinado mês. Outro exemplo, na tabela FACTO_INDEMNIZACAO_INV consegue-se saber qual é o valor total das indemnizações em pagamento num determinado mês. Tudo isto é difícil de determinar a partir das tabelas de factos transacionais porque para se saber a situação de algo num determinado momento no tempo, ter-se-ia de percorrer as transações desde o início da sua história.

A única diferença relativamente às dimensões destas tabelas comparadas com as dimensões das tabelas de factos transacionais, descritas na Secção 6.1, está na existência da dimensão MÊS, que é um subconjunto da dimensão DATA. Os factos destas tabelas já foram descritos na Secção 5.5. Os diagramas dos esquemas em estrela destes quatro processos de negócio encontram-se no Apêndice B.

6.3 Tabelas Agregadas

Tabelas agregadas são tabelas de factos físicas num DW que derivam das tabelas de factos no nível de detalhe mais alto, contendo factos agregados através de funções de soma, contagens, máximos, mínimos, etc. Quando criamos estas tabelas, estamos a eliminar dimensionalidade ou a associar os factos a dimensões num nível superior de sumarização. De forma a se obter um melhor desempenho, estas tabelas são usadas como substitutas sempre que a consulta o permite. Consultar uma tabela agregada é muito mais rápido porque tem menos registos, que uma tabela de factos no nível de detalhe mais alto, principalmente quando as dimensões associadas também são grandes. Assim, devido a esta vantagem, criámos tabelas agregadas no DW.

Primeiro tivemos que pensar que agregações poderiam ser úteis, ou seja, que tabelas agregadas é que seriam muitas vezes acedidas por consultas/relatórios e que ao mesmo tempo aumentassem o desempenho significativamente. Ao analisar as métricas e os relatórios definidos no levantamento de requisitos, rapidamente percebemos qual é a agregação necessária para quase todos os processos de negócio. Para as tabelas FACTO_APOLICE, FACTO_APOLICE_INV, FACTO_RECIBO, FACTO_RECIBO_INV, FACTO_SINISTRO, FACTO_SINISTRO_INV, pode ser feita uma agregação muito útil, simplesmente retirando a dimensão cobertura, porque muitas das análises que vão ser feitas pelos utilizadores não necessitam da dimensão cobertura. Por exemplo, o relatório de estatística de apólices, vai conter apenas métricas de contagem do número de apólices e, por isso, não precisa desta dimensão, porque não vai contar as coberturas de cada apólice.

Nas tabelas de factos FACTO_INDEMNIZACAO e FACTO_INDEMNIZACAO_INV esta agregação já não seria muito útil, pois raramente é criada uma mesma indemnização para duas coberturas e, por isso, a diferença entre o número de registo não seria muito significativa. Assim, não fizemos nenhuma agregação para estas duas tabelas porque não devemos criar tabelas agregadas que vão ter pouca utilidade e praticamente só vão ocupar espaço em disco desnecessariamente. Na Figura 6.5 podemos ver um dos esquemas em estrela sem a dimensão cobertura.

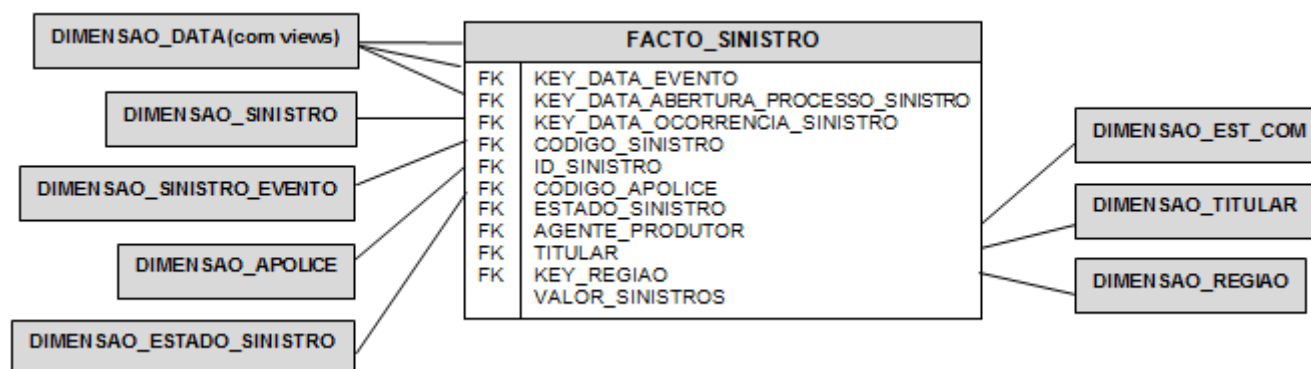


Figura 6.5: Tabela de Factos de Sinistro sem a Dimensão COBERTURA

Na Tabela 6.1 temos uma comparação entre o número de registos das tabelas de factos no nível de detalhe mais alto e das tabelas de factos agregados sem a dimensão COBERTURA. Como podemos perceber as consultas que não precisem da dimensão COBERTURA, vão ter que "percorrer" muito menos registos nas tabelas agregadas.

Tabelas de factos no nível de detalhe mais alto	Nº de registos	Tabelas de factos agregados sem a dimensão COBERTURA	Nº de registos
FACTO.APOLICE	12537	FACTO.APOLICE_SEMCOB	1856
FACTO.APOLICE_INV	102564	FACTO.APOLICE_SEMCOB_INV	11143
FACTO.RECIBO	15565	FACTO.RECIBO_SEMCOB	2024
FACTO.RECIBO_INV	779	FACTO.RECIBO_SEMCOB_INV	123
FACTO.SINISTRO	2616	FACTO.SINISTRO_SEMCOB	2075
FACTO.SINISTRO_INV	61	FACTO.SINISTRO_SEMCOB_INV	40

Tabela 6.1: Comparação entre o número de registos das tabelas de factos no nível de detalhe mais alto e das tabelas de factos agregados sem a dimensão COBERTURA

É de notar que este projeto foi desenvolvido num ambiente de testes e, por isso, o número de registos, que as tabelas do DW têm, é muito reduzido. No entanto, dá uma ideia próxima do que acontece na prática quando se compara o número de registos das tabelas de factos agregados com as de factos não agregados.

Algumas plataformas BI, como o MSTR, calculam o peso/tamanho lógico das tabelas. Este tamanho representa a granularidade de uma tabela e não se baseia no número de linhas da tabela. Este tamanho lógico é atribuído às tabelas baseado num algoritmo que usa o número de dimensões incluídas na tabela e o nível em que os atributos mapeados à tabela existem nas suas respetivas hierarquias. Este tamanho é utilizado quando temos tabelas de factos agregadas no nosso DW. A plataforma BI vai escolher sempre primeiro uma tabela com o tamanho lógico mais pequeno se a consulta o permitir e, assim, melhora o desempenho na execução dos relatórios. Esta funcionalidade é chamada de *aggregate navigator*.

6.4 Tabelas de Dimensão

Nas Figuras 6.6 - 6.20 podemos ver as dimensões presentes no DW e os seus atributos. Todas estas dimensões são partilhadas por tabelas de factos, como vimos na Figura 5.9.

DIMENSAO_DATA		
PK	KEY_DATA	DECIMAL(8, 0)
	DIA	DATE
	ANO	DECIMAL(4, 0)
	MES	DECIMAL(6, 0)
	MES_DESC	NCHAR(30)
	MES_DO_ANO	DECIMAL(2, 0)
	MES_DO_ANO_DESC	NCHAR(30)
	TRIMESTRE	DECIMAL(5, 0)
	TRIMESTRE_DESC	NCHAR(30)

Figura 6.6: Tabela de Dimensão Data

DIMENSAO_MES		
PK	KEY_MES	DECIMAL(6, 0)
	MES_DESC	NCHAR(30)
	MES_DO_ANO	DECIMAL(2, 0)
	MES_DO_ANO_DESC	NCHAR(30)
	TRIMESTRE	DECIMAL(5, 0)
	TRIMESTRE_DESC	NCHAR(30)
	ANO	DECIMAL(4, 0)

Figura 6.7: Tabela de Dimensão Mês

As Figuras 6.6 e 6.7 representam as tabelas DIMENSAO_DATA e DIMENSAO_MES. A dimensão MÊS é um subconjunto da dimensão DATA. Contudo foi necessário criar uma tabela física para esta dimensão, como foi referido na secção 5.5. No processo de carregamento, a tabela DIMENSAO_MES é carregada a partir da tabela DIMENSAO_DATA para garantir a conformidade entre as tabelas. A chave da tabela DIMENSAO_MES é carregada com os valores únicos do campo MES da tabela DIMENSAO_DATA.

Na dimensão DATA o nível Dia contém os atributos KEY_DATA (código) e DIA (data do calendário). O nível Mês desta dimensão é composto pelos atributos MES (número do mês e do ano, ex: 062013) e MES_DESC (descrição do mês e do ano, ex: "Junho de 2013"). O nível Mês do Ano é composto pelos atributos MES_DO_ANO (número do mês no ano, ex: 06) e MES_DO_ANO_DESC (descrição do mês, ex: "Junho").

DIMENSAO_REGIAO		
PK	KEY_REGIAO	BIGINT
	ID_DISTRITO	NCHAR(30)
	DISTRITO	NCHAR(30)
	ID_CONCELHO	NVARCHAR(11)
	CONCELHO	NVARCHAR(2)
	CODIGO_POSTAL	NVARCHAR(5)

Figura 6.8: Tabela de Dimensão Região

DIMENSAO_APOLICE		
PK	CODIGO_APOLICE	DECIMAL(10, 0)
	TIPO_SEGURO	DECIMAL(2, 0)
	PRODUTO	DECIMAL(4, 0)
	PRODUTO_DESC	NCHAR(60)
	RAMO	DECIMAL(2, 0)
	RAMO_DESC	NCHAR(60)
	APOLICE_ANUAL	NCHAR(1)
	APOLICE_TEMPORARIA	NCHAR(1)

Figura 6.9: Tabela de Dimensão Apólice

DIMENSAO_APOLICE_EVENTO		
PK	ID_APOLICE	DECIMAL(15, 0)
	RAZAO_ANULACAO_APOLICE	DECIMAL(2, 0)
	RAZAO_ANULACAO_APOLICE_DESC	NCHAR(60)
	SUBRAZAO_ANULACAO_APOLICE	DECIMAL(2, 0)
	SUBRAZAO_ANULACAO_APOLICE_DESC	NCHAR(60)

Figura 6.10: Tabela de Dimensão Apólice Evento

DIMENSAO_RECIBO		
PK	CODIGO_RECIBO	DECIMAL(15, 0)
	TIPO_RECIBO	NCHAR(2)
	ESPECIE_RECIBO_REC	NVARCHAR(1)
	ESPECIE_RECIBO_REC_DESC	NCHAR(60)
	TIPO_MULTI	NVARCHAR(2)
	TIPO_MULTI_DESC	NCHAR(30)

Figura 6.11: Tabela de Dimensão Recibo

DIMENSAO_RECIBO_EVENTO		
PK	ID_RECIBO	DECIMAL(15, 0)
	CANAL_COBRANCA	NCHAR(2)
	CANAL_COBRANCA_DESC	NCHAR(60)
	COBRADOR_REC	DECIMAL(8, 0)
	COBRADOR_REC_DESC	NCHAR(60)
	TIPO_COBRANCA	NVARCHAR(1)
	TIPO_COBRANCA_DESC	NCHAR(30)

Figura 6.12: Tabela de Dimensão Recibo Evento

Quase todos os níveis das dimensões, identificados na Secção 5.3, são compostos apenas por um ou dois atributos. Os níveis com dois atributos contêm simplesmente um código e uma descrição. Por exemplo, na tabela DIMENSAO_APOLICE, os dois atributos RAMO e RAMO_DESC, do nível Ramo, são o código e a descrição do ramo, respetivamente. A maior parte dos códigos são apenas utilizados como "ID" na criação dos atributos na camada lógica no MicroStrategy e somente as descrições aparecem nos relatórios. No entanto, os utilizadores também podem querer ver os códigos de alguns dos atributos nos relatórios, como é o caso do atributo COBERTURA (Figura 6.17).

DIMENSAO_SINISTRO		
PK	CODIGO_SINISTRO	NVARCHAR(34)
	TIPO_DANO_SINISTRO	NCHAR(34)
	ANO_OCORRENCIA	NCHAR(34)
	TIPO_IDS_CIDS	NCHAR(34)
	TIPO_IDS_CIDS_DESC	NCHAR(60)
	AJUSTE_REEMBOLSO	NCHAR(34)

Figura 6.13: Tabela de Dimensão Sinistro

DIMENSAO_SINISTRO_EVENTO		
PK	ID_SINISTRO	DECIMAL(15,0)
	GESTOR_SINISTRO	NVARCHAR(34)
	GESTOR_SINISTRO_DESC	NCHAR(60)
	EQUIPA_GESTOR_SINISTRO	NVARCHAR(34)
	EQUIPA_GESTOR_SINISTRO_DESC	NCHAR(60)
	ESTADO_VITIMA	NCHAR(34)
	SITUACAO_SINISTRO	DECIMAL(34, 0)
	SITUACAO_SINISTRO_DESC	NCHAR(60)
	ACAO_JUDICIAL_SINISTRO	NCHAR(34)
	AREA_GESTAO	NCHAR(34)
	AREA_GESTAO_DESC	NCHAR(60)
	REABERTURA_SINISTRO	NVARCHAR(34)

Figura 6.14: Tabela de Dimensão Sinistro Evento

DIMENSAO_INDEMNIZACAO		
PK	CODIGO_INDEMNIZACAO	NCHAR(15)
	PRAZO_PAGAMENTO	DECIMAL(11, 0)
	ESPECIE_RECIBO_IND	NCHAR(1)

Figura 6.15: Tabela de Dimensão Indemnização/Reembolso

DIMENSAO_INDEMNIZACAO_EVENTO		
PK	ID_INDEMNIZACAO	DECIMAL(15, 0)
	ENTIDADE_RECEBEDORA	DECIMAL(11, 0)
	ENTIDADE_RECEBEDORA_DESC	NCHAR(60)
	TIPO_ENTIDADE_RECEBEDORA	NVARCHAR(3)
	TIPO_ENTIDADE_RECEBEDORA_DESC	NCHAR(60)
	COBRADOR_IND	DECIMAL(8, 0)
	COBRADOR_IND_DESC	NCHAR(60)
	FORMA_COBRANCA	NVARCHAR(1)
	FORMA_COBRANCA_DESC	NCHAR(30)

Figura 6.16: Tabela de Dimensão Indemnização/Reembolso Evento

DIMENSAO_COBERTURA		
PK	KEY_COBERTURA	BIGINT
	COBERTURA	DECIMAL(7, 0)
	COBERTURA_DESC	NCHAR(60)
	RAMO_CONTABILISTICO	NVARCHAR(10)
	RAMO_CONTABILISTICO_DESC	NCHAR(60)

Figura 6.17: Tabela de Dimensão Cobertura

DIMENSAO_ESTADO_X		
PK	X_ESTADO	NVARCHAR(11)
	X_ESTADO_DESC	NVARCHAR(200)

Figura 6.18: Tabela de Dimensão Estado X, onde $X = \{APOLICE, RECIBO, SINISTRO, IND.REEMB\}$

As dimensões na Figura 6.18 possuem apenas um nível, como vimos na Secção 5.3, composto apenas por dois atributos: um que contém os códigos e o outro as descrições dos estados. Por exemplo, considerando a tabela de FACTO_APOLICE que está mapeada com a dimensão ESTADO_APOLICE através do atributo APOLICE_ESTADO, este atributo indica se um registo é de uma apólice nova, anulada, reposta em vigor, etc. Se considerarmos a tabela de factos FACTO_SINISTRO que está mapeada com a dimensão ESTADO_SINISTRO através do atributo SINISTRO_ESTADO, este atributo indica se a transação de um sinistro é uma abertura, fecho, etc. Se considerarmos o atributo ESTADO_IND_REEMB incluído na tabela de factos FACTO_INDEMNIZACAO, podemos perceber se a transação é de uma indemnização ou de um reembolso e ainda se é, por exemplo, uma anulação ou uma cobrança. Estes atributos vão ser mais utilizados nos filtros porque permitem implementar métricas só para apólices/sinistros/etc que estão num determinado estado. Por exemplo, com a dimensão ESTADO_APOLICE, podemos implementar uma métrica que calcule o valor do prémio das apólices novas e outra que calcule o valor do prémio das apólices anuladas.

DIMENSAO_TITULAR		
PK	TITULAR	DECIMAL(8, 0)
	TITULAR_DESC	NVARCHAR(60)
	CAE	DECIMAL(6, 0)
	SEXO	NVARCHAR(1)
	TELEFONE	NVARCHAR(13)
	TELEMOVEL	NVARCHAR(9)
	ESTADO_CIVIL	NVARCHAR(1)
	TIPO_EMPRESA	NVARCHAR(60)
	EMAIL	NVARCHAR(60)
	NIF	DECIMAL(9, 0)

Figura 6.19: Tabela de Dimensão Titular

DIMENSAO_EST_COM		
PK	AGENTE_PRODUTOR	DECIMAL(8, 0)
	AGENTE_DESC	NVARCHAR(60)
	TECNICO_COMERCIAL	DECIMAL(8, 0)
	TECNICO_COMERCIAL_DESC	NVARCHAR(60)
	ESCRITORIO_LOCAL	DECIMAL(8, 0)
	ESCRITORIO_LOCAL_DESC	NVARCHAR(60)
	ESCRITORIO_REGIONAL	DECIMAL(8, 0)
	ESCRITORIO_REGIONAL_DESC	NVARCHAR(60)
	DIRECAO_REGIONAL	DECIMAL(8, 0)
	DIRECAO_REGIONAL_DESC	NVARCHAR(60)
	DIRECAO_NACIONAL	DECIMAL(8, 0)
	DIRECAO_NACIONAL_DESC	NVARCHAR(60)
	COMPANHIA	DECIMAL(8, 0)
	COMPANHIA_DESC	NVARCHAR(60)

Figura 6.20: Tabela de Dimensão Estrutura Comercial

No MicroStrategy, para ser possível analisar os dados através do *drill down* e do *roll up*, tivemos que indicar no MSTR Architect como são todas as hierarquias destas dimensões, ilustradas na Secção 5.3.

6.5 Tabela de Objetivos e KPIs

No DW além de todas as tabelas descritas até agora, também foi carregado uma tabela de objetivos. Estas tabelas de objetivos são tabelas inseridas pelos próprios clientes na base de dados e contêm os objetivos de negócio. A tabela de objetivos é uma tabela de factos e as suas medidas (factos) vão ser utilizadas para criar os indicadores chave de desempenho (KPIs). A Figura 6.21 ilustra a tabela de factos de objetivos que foi carregada no DW.

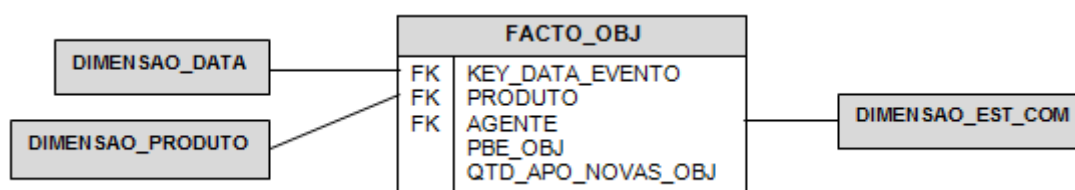


Figura 6.21: Tabela de Factos de Objetivos

Com esta tabela os utilizadores podem criar dois KPIs: um indica o objetivo a atingir relativamente ao número de apólices novas e o outro indica o objetivo para o prémio bruto emitido. Com esta tabela, os utilizadores definem estes objetivos por agente, produto e data. Estes KPIs vão depois ser comparados nos *scorecards* com as métricas correspondentes calculadas a partir das outras tabelas de factos existentes. Nos *scorecards* os utilizadores vão conseguir ver, por exemplo, se os agentes atingiram ou não os objetivos propostos em relação ao número de apólices vendidas, comparando o KPI criado a partir da tabela de objetivos com a métrica que calcula a quantidade de apólices novas.

Os KPIs são criados a partir de tabelas de sumarização periódica porque são métricas que utilizam medidas instantâneas que se referem ao final de um período de tempo. Como a tabela de objetivo é uma tabela de factos de sumarização periódica, quando tentámos criar *scorecards* com comparações entre os KPIs e as métricas que utilizavam medidas das tabelas de factos transacionais, deparámo-nos com um problema: os *scorecards* estavam a ficar extremamente complexos, envolvendo várias prompts e vários atributos de data, só para medir algo num determinado intervalo de tempo e comparar com os KPIs.

A solução foi criar duas tabelas de factos de sumarização periódica a partir das tabelas transacionais de apólice e recibo, para poder comparar as suas métricas com os KPI, de uma forma simples e eficiente. Nas Figuras 6.22 e 6.23 podemos ver os esquemas em estrela que incluem essas duas tabelas.

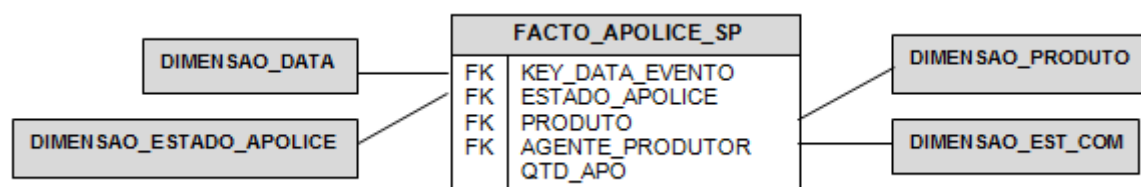


Figura 6.22: Tabela de Factos de Sumarização Periódica de Apólice

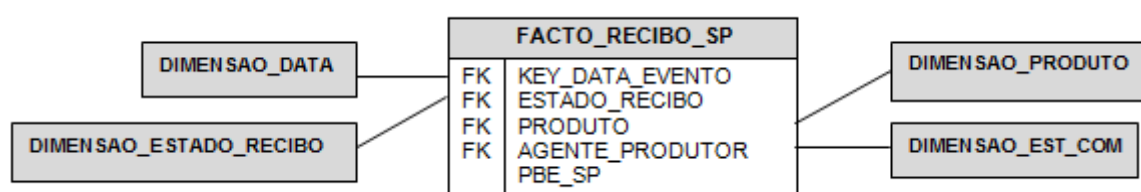


Figura 6.23: Tabela de Factos de Sumarização Periódica de Recibo

Apesar de serem de sumarização periódica, estas tabelas são diferentes das de inventário. Ao contrário das tabelas de inventário que contêm medidas instantâneas no final de um período (mês), estas tabelas contêm medidas acumuladas durante um determinado período de tempo (ano). Desta forma, é possível ver o desempenho acumulado do negócio num determinado intervalo de tempo. Estas tabelas são calculadas a partir das tabelas de factos transacionais, representando uma agregação da atividade transacional que ocorreu durante o ano, e, para já, são então apenas necessárias por razões de desempenho e simplicidade na construção e execução de *scorecards*. A granularidade destas tabelas é uma linha por produto e agente produtor a cada dia. Como estas tabelas estão num nível de sumarização elevado, esta acumulação pode ser feita por dia que a tabela não fica muito grande. Quando for necessário diminuir o nível de granularidade, então aí deve-se repensar se devemos ou não fazer um *roll up* na dimensão DATA para mês. A Figura 6.25 mostra um relatório muito simples que contém uma métrica que indica o número de apólices vendidas por cada um dos agentes por produto e um KPI que indica qual era/é o respetivo objetivo a atingir. Com estes relatórios consegue-se avaliar a prestação de cada um dos agentes produtores e perceber em que produtos existe uma maior facilidade/dificuldade de venda.

Objetos do relatório

Nome	Tipo
Agente Produtor	Atributo
Ano	Atributo
Produto	Atributo
Classificação	Métrica
Diferença	Métrica
Quantidade de apólices novas (objetivo)	Métrica
Quantidade de apólices novas (sp)	Métrica

Navegador de Objetos

Explorador de dados - Projeto

Nome	Tipo
Apólice	Hierarquia
Estrutura Comercial	Hierarquia
Geografia	Hierarquia
Indemnização	Hierarquia
Ramo Contabilístico até Cobertura	Hierarquia
Sinistro	Hierarquia
Tempo	Hierarquia
Titular	Hierarquia
Hierarquia de Sistema	Hierarquia do Sistema

Detalhes do Relatório

Filtro de exibição: Novo Limpar

Ano: Não relacionado {2012}

		Ano	2013			
		Métricas	Quantidade de apólices novas (sp)	Quantidade de apólices novas (objetivo)	Classificação	Diferença
Agente Produtor	Produto					
JOSÉ ANTUNES	I2S AUTO	2620	2	10	■	(8)
	AC.TRABALHO-C. OUTRÉM-PR.FIXO	2001	19	20	■	(1)
MÁRIO ANDRÉ GOMES	M.R.H-CONF.TOTAL/EXCL	2306	27	25	■	2
	I2S AUTO	2620	71	30	■	41
JORGE FREITAS LEMOS	M.R.H-CONF.TOTAL/EXCL	2306	4	12	■	(8)
	I2S CASA	3501	2	50	■	(48)
	AT FIXO	1001	7	10	■	(3)
INÊS SILVA PEREIRA	AC.TRABALHO-C. OUTRÉM-PR.FIXO	2001	2	50	■	(48)
	I2S AUTO	2620	3	10	■	(7)

Figura 6.24: Exemplo de Relatório de Objetivos

6.6 Esquema em Floco de Neve

No desenho e desenvolvimento de um DW, devemos sempre utilizar o esquema em estrela, porque obtemos um melhor desempenho nas consultas e é mais simples que o esquema em floco de neve, como já referi na Subsecção 2.2.4. No entanto, o esquema em estrela também tem as suas desvantagens e uma delas é a inflexibilidade. Com este esquema, não conseguimos retirar um subconjunto de atributos de uma dimensão e "oferecê-los" a uma tabela de factos. Por exemplo, quando temos uma tabela de factos que quer aceder a um nível superior de sumarização de uma dimensão, no esquema em estrela, essa tabela de factos não pode aceder à tabela base dessa dimensão e seleccionar o subconjunto de atributos que se aplicam a esse nível mais elevado de sumarização. Como vimos na Secção 5.4, para a dimensão DATA e para as tabelas de factos de inventário, a única solução é criar outra tabela de dimensão com esse subconjunto de atributos necessário, que, naquele caso em particular, foi a tabela DIMENSAO_MES.

Assim, para evitar esta inflexibilidade do esquema em estrela, podemos optar pelo esquema em floco de neve. Neste esquema, a dimensão é normalizada e, assim, uma tabela de factos pode aceder aos atributos que quiser.

Além do caso descrito na Secção 5.4, esta situação também ocorreu na Secção 6.4. Como se pode reparar, as tabelas de factos de sumarização periódica estão ligadas a uma dimensão PRODUTO. Esta dimensão é simplesmente a tabela DIMENSAO_APOLICE num nível superior de sumarização, ou seja, é um subconjunto da dimensão APÓLICE. Como essas tabelas de factos

estão num nível de granularidade superior às restantes tabelas de factos e querem aceder apenas a um subconjunto da dimensão APÓLICE, então, como é óbvio, não se podem ligar à mesma tabela de dimensão. Uma das soluções é então normalizar a tabela DIMENSAO_APOLICE para poder ser acedida por várias tabelas de factos com níveis de granularidade diferentes. A Figura 6.25 ilustra a normalização da tabela DIMENSAO_APOLICE. O esquema em floco de neve foi muito recomendado por um consultor da MicroStrategy, que afirmou que a ferramenta de BI MSTR também é compatível com este esquema. Outra solução para esta situação seria a optada para a dimensão DATA. Teríamos de criar uma tabela de dimensão PRODUTO, mas a dimensão APÓLICE continuava a ter os atributos de PRODUTO.

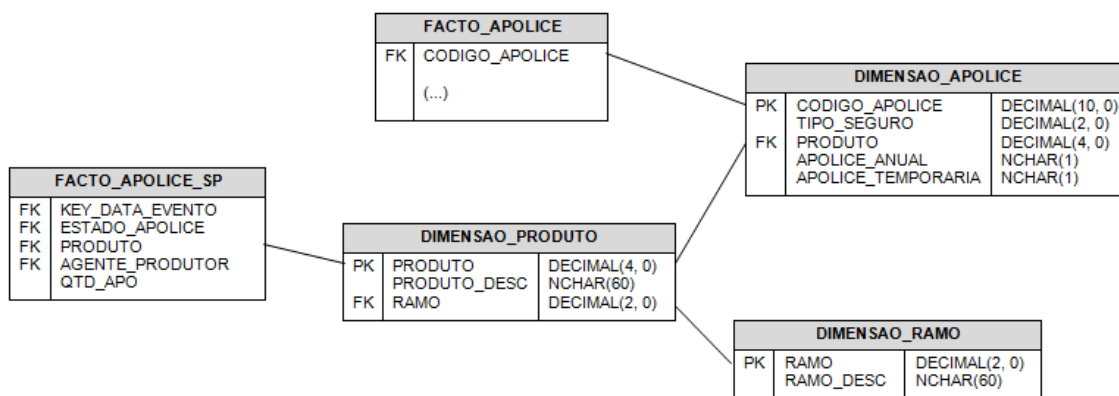


Figura 6.25: Normalização da Tabela de Dimensão Apólise

Capítulo 7

Desenvolvimento do Processo de Carregamento

Na realização dos processos de integração de dados no DW, decidiu-se, para este projeto, utilizar ao máximo o software Talend Open Studio for Data Integration. Esta ferramenta organiza todos os passos necessários para executar um processo de ETL em *Jobs*. Dentro de cada Job existem componentes que fazem a extração, as transformações e o carregamento dos dados, permitindo assim realizar as três fases de um processo ETL numa só tarefa.

Como já referi na Subsecção 3.2.1, optámos por utilizar a versão *freeware* do Talend e não adquirir a versão *comercial*, o Talend Enterprise Data Integration. Isto traz algumas desvantagens e, por isso, tivemos que arranjar alternativas para algumas tarefas que só com a versão comercial do Talend seriam possíveis de realizar.

A primeira tarefa é a programação da execução dos *Jobs* que não é possível na versão *freeware*. O processo de integração está programado para correr todas as madrugadas, mas para isso temos que utilizar um *scheduler* fora do Talend. Isto não seria possível, sem a funcionalidade de exportar os *Jobs* existente no Talend, que permite exportá-los e depois executá-los independentemente do Talend e em qualquer servidor. Esta funcionalidade exporta todos os ficheiros necessários para executar o *Job*, incluindo o *.bat* e o *.sh*. Todos os *jars* necessários são incluídos nos comandos *.bat* ou *.sh*.

A segunda limitação é não podermos configurar os *Jobs* para efetuarem um carregamento incremental. Nesta versão, sempre que executamos os *Jobs*, os dados são todos extraídos da fonte e processados independentemente se são dados novos ou não e, por isso, é sempre executado um carregamento total dos dados. Assim, tivemos que implementar um processo fora do Talend, que nos dias seguintes à migração inicial, carregue apenas os registos inseridos ou atualizados desde o último carregamento. Além disso, também se tem que passar para o lado do DW uma indicação de quais os registos que foram apagados. Para ser possível fazer o carregamento incremental usamos *triggers* no Informacional e tabelas que armazenam *flags* e *timestamps*.

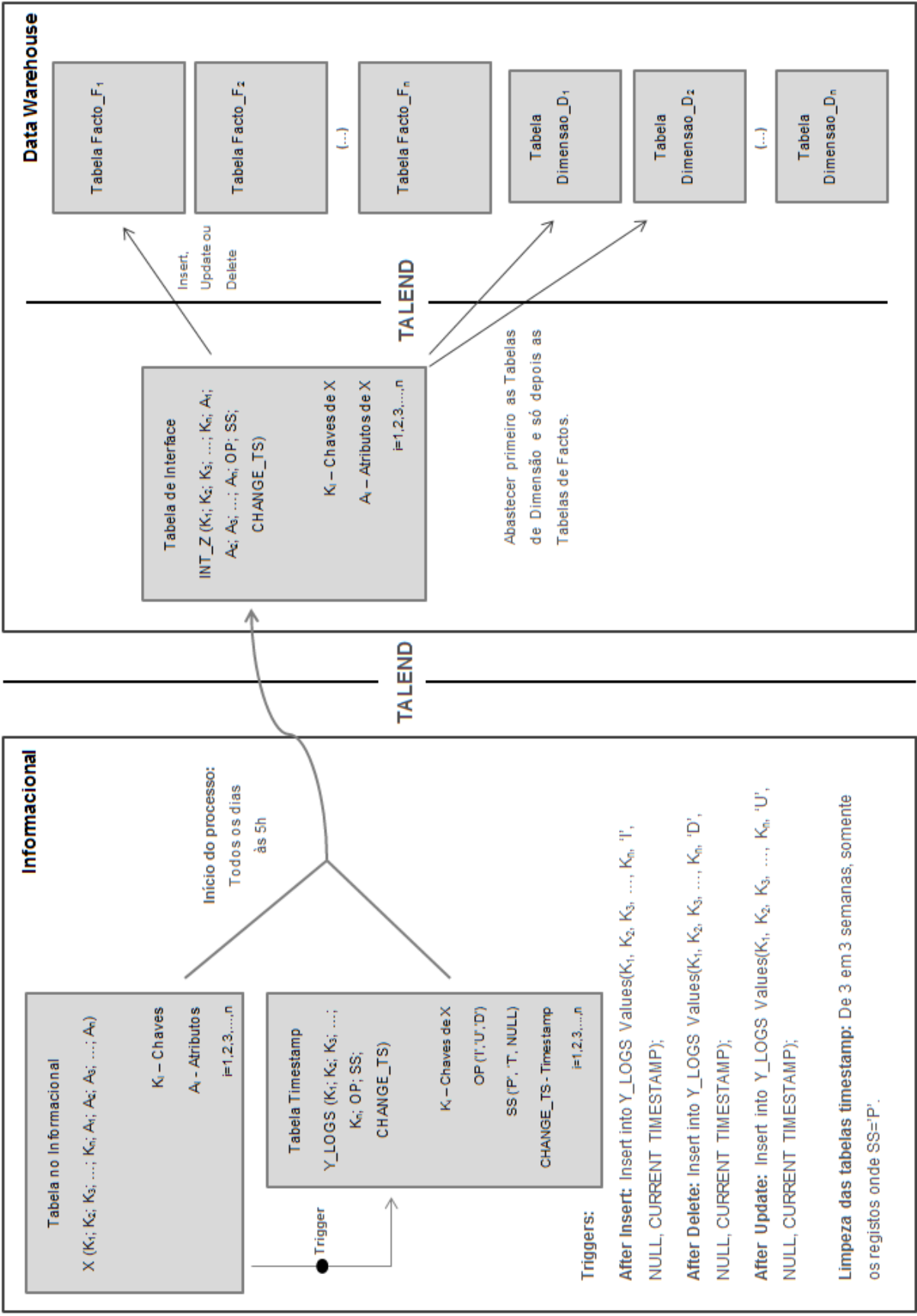


Figura 7.1: Diagrama do processo de abastecimento incremental do DW

Na Figura 7.1, temos um esquema do processo de carregamento de registos que ainda não existem no DW. Como exemplo, neste esquema, podemos tomar $X=R\#MOPAPO$ e $Y=APOLICE$. Neste processo, quando é, por exemplo, inserido um novo registo na tabela $R\#MOPAPO$ do Informacional, é "ativado" um *trigger* que insere um registo na tabela Timestamp (designada $APOLICE_LOGS$) com os valores das chaves da tabela $R\#MOPAPO$ referentes ao registo inserido. Além das chaves também são inseridos:

- um campo OP que indica se a operação foi um *Insert*, um *Update* ou um *Delete* (neste caso seria $OP="I"$),
- um campo SS que toma o valor "P" se o registo já tiver sido processado e já existe no DW, o valor "T" se o registo estiver a ser carregado no DW, ou o valor "NULL" se as duas situações anteriores não se confirmarem (neste caso seria o valor "NULL"), e
- um campo $CHANGE_TS$ com o *timestamp* atual.

Desta forma, quando fazemos uma junção entre a tabela do Informacional e tabela Timestamp só carregamos para o lado do DW os registos novos.

Quando os registos chegam ao DW são armazenados em tabelas de interface e só depois é que os dados são carregados nas tabelas do modelo dimensional. Como referimos, no DW não devem existir tabelas que não servem para ser consultadas, mas, neste caso, como não se usou uma área de estágio, esta foi a melhor solução encontrada para simular essa área. Se não utilizássemos estas tabelas de interface, as operações de junção entre tabelas, as transformações e a criação de tabelas agregadas iriam ser realizadas no momento da passagem entre a máquina onde está alojado o Informacional e a máquina onde está alojado o DW. Isto poderia trazer um problema quando o processo de carregamento falhasse ou quando as comunicações fossem abaixo porque teria que se extrair novamente os dados do Informacional. Assim, a solução foi fazer um carregamento mais direto possível entre as duas máquinas. Estas tabelas de interface basicamente vão ser cópias das tabelas do Informacional descritas na Secção 4.2, com mais os campos presentes na tabela Timestamp. Só depois destas tabelas serem carregadas com dados é que abastecemos as tabelas de factos e dimensões. Como já não há dependência de comunicações, este processo já pode ser mais lento.

No Talend, entre as duas máquinas, vai existir a sequência de processos seguinte:

1. Alterar todos os valores null na coluna SS da tabela $Time\Stamp$ para "T".
2. Fazer a junção entre a tabela X e a tabela X_LOGS , onde o campo SS é igual a "T", e carregar os dados resultantes nas tabelas de interface.
3. Alterar todos os valores "T" na coluna SS da tabela $Time\Stamp$ para "P".

As tabelas de Timestamp não vão ser sempre limpas no final deste processo, para não sobrecarregar o processo com mais uma tarefa. Pode-se limpar estas tabelas por exemplo de três em três semanas para não ficarem muito grandes, e somente os registos onde SS="P". Assim, temos um registo daquilo que foi feito no DW nos últimos tempos caso seja preciso.

No data warehouse vão existir tabelas de dimensão com poucos registos e, por isso, o carregamento de dados nestas tabelas pode ser total, porque é rápido. Assim, no segundo passo, também são carregados os registos diretamente nas tabelas de dimensão mais pequenas, não passando pelas tabelas de interface. No segundo passo, também é carregada uma tabela de interface (migração total) que junta todas as descrições necessárias para popular as tabelas de dimensão que são carregadas através o processo incremental. Esta tabela de interface chama-se INT_MOPDIMDSG.

Nas próximas três secções, serão explicados três processos de carregamento. Serão apenas descritos três processos, porque os *Jobs* que carregam as restantes tabelas de dimensão e factos não têm muitas diferenças comparados com os que irão ser explicados. Na quarta e última secção irá ser descrito um processo de carregamento que recorre a um stored procedure no SQL Server 2008.

7.1 Carregamento Total da Tabela de Dimensão de Região

Um exemplo de uma tabela de dimensão que é carregada totalmente todos os dias, sem passar pelas tabelas de interface, é a DIMENSAO_REGIAO. A Figura 7.2 mostra o *Job* implementado no Talend.

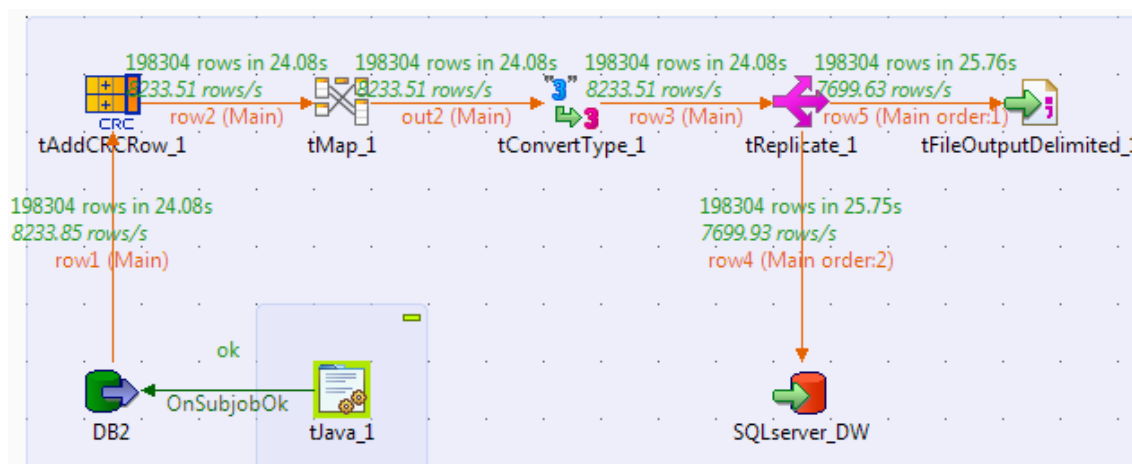


Figura 7.2: Carregamento Total da Tabela de Dimensão REGIAO

O componente "DB2" extrai os valores das tabelas que contém os atributos necessários para passar para a tabela DIMENSAO_REGIAO. O componente seguinte, chama-se tAddCRCRow e gera uma chave substituta (*surrogate key*). Estas chaves substitutas, habitualmente inteiros,

são geradas sequencialmente neste componente para popular as tabelas de dimensão. Estas chaves são utilizadas como chaves primárias nas tabelas de dimensão e não têm significado de negócio. Apenas são utilizadas nas operações de junção entre as tabelas de dimensão e as tabelas de factos. Ralph Kimball defende a utilização destas chaves em todas as tabelas de dimensão, porque podem ocorrer alterações no DW sobre os códigos operacionais de produção utilizados como chaves primárias [20]. Existe depois o componente tMap que serve para filtrar colunas e linhas, para fazer uma série de transformações, junções entre tabelas e permite múltiplos *outputs*. Neste caso, o tMap está a filtrar colunas e a alterar os nomes das colunas. O componente seguinte, chama-se tConvertType e converte alguns tipos de campos (Java) para outros, por exemplo *bigdecimal* para *string*. A seguir o fluxo é duplicado pelo componente tReplicate, permitindo a passagem para dois componentes que se chamam tFileOutputDelimited e SQLserver_DW. No componente tFileOutputDelimited os registos são escritos num ficheiro. Este ficheiro pode servir como *backup* ou então para ser chamado por outros *Jobs*, evitando um acesso à tabela de dimensão no DW. Neste caso, vai ser chamado pelos *Jobs* das tabelas de factos para inserir a chave substituta da REGIÃO na tabela. O outro componente SQLserver_DW limpa e abastece a tabela de destino com os novos dados. Como podemos verificar, este processo demora cerca de 26 segundos a carregar cerca de 200.000 registos.

7.2 Carregamento da Tabela de Dimensão de Titular

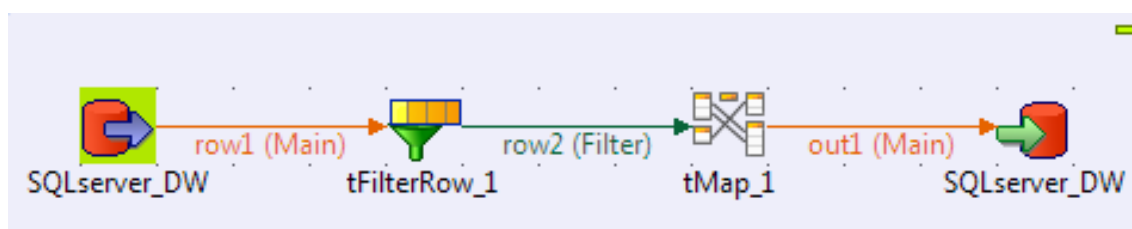


Figura 7.3: Carregamento Incremental da Tabela de Dimensão TITULAR

Na Figura 7.3, temos um exemplo de uma tabela de dimensão que é carregada de forma incremental porque pode ter centenas de milhar ou até milhões de registos - a tabela DIMENSAO_TITULAR. Depois de carregadas as tabelas de interface, o Job executado para abastecer esta tabela, recebe todos os campos distintos necessários para abastecer a tabela de interface das apólices. Todos os atributos necessários para abastecer a tabela DIMENSAO_TITULAR, existem nesta tabela de interface. Podemos ir buscar a informação sobre os titulares só a esta tabela porque normalmente nas apólices são registados sempre os titulares e as suas informações, não fazendo sentido existir um titular, por exemplo, na tabela de sinistro que não existe na tabela de apólice. Neste *Job*, depois da extração de todas as linhas da tabela de interface de apólice, o processo segue para o componente tFilterRow que só deixa passar os registos que tenham OP igual a "U" ou "I" (atualizações e inserções). O próximo componente tMap filtra a coluna OP

e, no fim, os registos são inseridos ou atualizados (existe uma opção que permite escolher estas duas operações) na tabela DIMENSAO_TITULAR. Nos *Jobs* implementados para abastecer as tabelas de dimensão, os registos não são apagados, porque os *Jobs* iriam ficar demasiado complexos e lentos porque teria-se-ia que verificar a inexistência de outros registos nas tabelas de factos a apontar para a mesma chave primária na tabela de dimensão. No entanto, isso não é problema, porque como não há nenhum registo na tabela de factos a apontar para esse valor, esse titular nunca aparecerá nos relatórios. No entanto, pode-se fazer um *full refresh* direto, sem passar pelas tabelas de interface, por exemplo, de seis em seis meses, para estes registos desaparecerem.

Normalmente os valores dos atributos nunca sofrem alterações ou são apagados. Contudo, como já vimos, neste DW, isso pode acontecer em qualquer uma das tabelas de dimensão, quando existem erros ou uma alteração natural no sistema operacional. Quando o valor de um atributo é modificado, o histórico nunca é mantido e, assim, perde-se o conhecimento do valor anterior. Este tipo de alterações são consideradas de tipo 1, segundo Carlos Caldeira, em [7].

7.3 Carregamento da Tabela de Factos das Apólices

Na Figura 7.4 temos o carregamento incremental da tabela FACTO_APOLICE. O *Job* começa por extrair todos os registos da tabela de interface apólice com o objetivo de abastecer a tabela de factos FACTO_APOLICE e a tabela agregada. Depois da extração, o fluxo de dados é duplicado e cada um passa por um componente tFilterRow. No fluxo de cima apenas passam as atualizações e as inserções e no de baixo, os *deletes*. Seguindo no fluxo de cima, o componente tUniqRow faz uma operação "DISTINCT" aos registos com chaves iguais (que só se diferenciam no campo "OP") de forma a ficar apenas a instrução mais recente de todas, retirando as instruções desatualizadas que podem acontecer relativamente a um registo. O próximo componente chama-se tMap que recebe de *input* os registos e dois ficheiros que contêm os valores de algumas tabelas de dimensão (entre elas a de DIMENSAO_REGIAO). No tMap vai haver uma junção entre estas "tabelas" e a tabela de interface, para, deste modo, se substituir os atributos pelas chaves substitutas que vão mapear a tabela de factos com as tabelas de dimensão.

No componente tMap também é testado se os registos que foram inseridos ou atualizados, não foram posteriormente apagados ao impedir a passagem dos registos que têm valores "NULL". Por fim, neste componente, algumas colunas que não interessam são filtradas (incluindo a coluna "OP"). Depois o fluxo é duplicado para inserir ou atualizar os registos na tabela de detalhe e na tabela agregada. Antes de chegar à tabela agregada, ainda passa por um componente que se chama tAggregateRow, que agrega os factos e agrupa-os por todos os atributos menos os que se encontram no nível da cobertura, como a própria chave substituta da cobertura. Desta forma, a informação é agregada para o nível de granularidade de um registo para cada transação de uma apólice. O componente tUniqRow é necessário para o

tAggregateRow não somar ou contar factos repetidos que pertencem a registos com chaves iguais. Por fim, os registos são inseridos ou atualizados na tabela agregada de apólice.

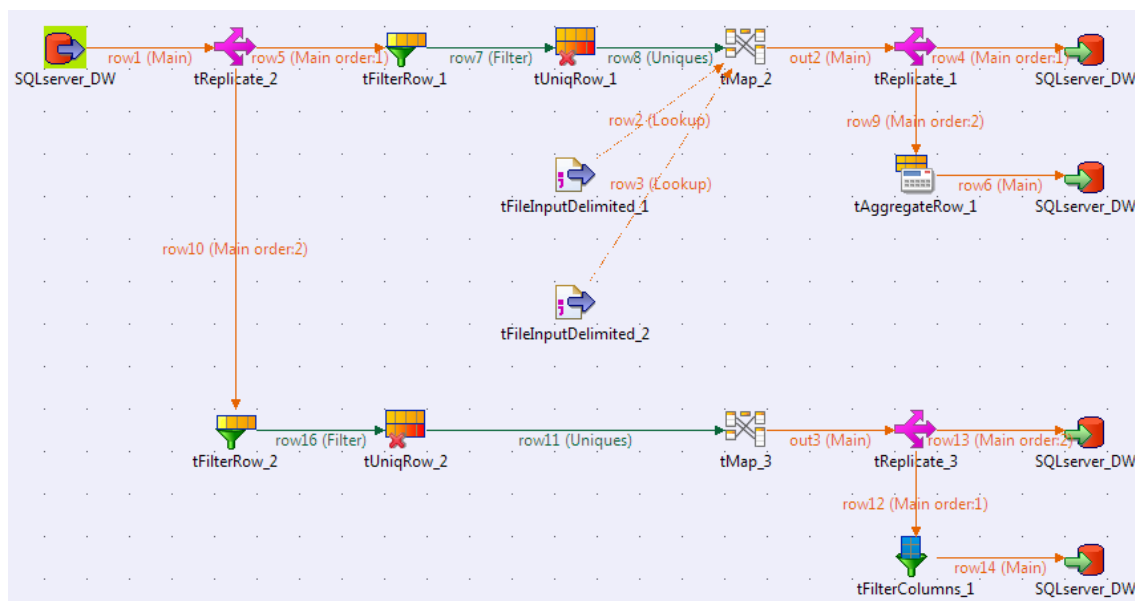


Figura 7.4: Carregamento Incremental da Tabela de Factos APÓLICE

No fluxo de baixo, os registos com OP="D" passam também por um componente tUniqRow que faz o "DISTINCT" aos registos que tenham as mesmas chaves, ficando apenas o último registo, o mais recente. Isto é necessário pois quando um registo é apagado da tabela de factos de apólices e a seguir vier uma operação para apagar o mesmo registo, o processo vai retornar erro, pois ele já não existe na tabela. A seguir vem o componente tMap que testa se os registos que foram apagados, não voltaram posteriormente a ser criados, ao impedir a passagem dos registos que têm valores (sem ser os campos chave nas tabelas de interface). Depois o fluxo é duplicado para apagar os registos na tabela de detalhe e na tabela agregada. Antes de chegar à tabela agregada os campos de risco são apagados.

7.4 Carregamento Total Mensal da Tabela de Dimensão de Estrutura Comercial

Muitas das tabelas de dimensão são abastecidas a partir das tabelas de interface de apólice, sinistro, etc, que por sua vez são abastecidas a partir das tabelas descritas na Secção 4.2. Por exemplo, como referido anteriormente, uma dessas tabelas é a DIMENSAO_TITULAR. No entanto, nem todas as tabelas de dimensão são carregadas a partir dessas tabelas. Por exemplo, a DIMENSAO_EST_COM é carregada a partir de uma tabela, também presente no Informacional, mas que contém apenas os atributos da estrutura comercial.

Essa tabela da estrutura comercial presente no Informacional é limpa e carregada no início de cada mês. Durante o resto do tempo, ela mantém-se intacta, porque as alterações da estrutura comercial, nas companhias de seguros, só entram em vigor no início de cada mês. Logo, a tabela de dimensão DIMENSAO_EST_COM também só é limpa e totalmente carregada uma vez por mês. Esse carregamento poderia ser direto, sem passar pelas tabelas de interface. Contudo, existem alguns fatores que impedem isso mesmo:

- o número de atributos nesta tabela varia conforme a companhia de seguros em questão;
- o desenho da tabela presente no Informacional é diferente e impede que seja realizado um carregamento direto para a tabela de dimensão do DW.

Assim, tem que existir uma tabela de interface, porque é necessário executar um processo complexo de transformação de dados, que não deve ser feito na passagem dos dados entre a máquina que aloja o Informacional e a máquina que aloja o DW, devido às razões já enumeradas anteriormente. Essa tabela de interface é uma réplica da tabela fonte presente no Informacional e chama-se INT_COM_STR_R.

	MEC_ANME	MEC_PROD	MEC_SEQP	MEC_DCNIV	MEC_ENTP
1	201308	103	0	COMPANHIA	11030
2	201308	103	1	DIR. NACIONAL	47001
3	201308	103	2	DIR. REGIONAL	46002
4	201308	103	3	ESCR. REGIONAL	46023
5	201308	103	4	ESCR.LOCAL	46231
6	201308	103	5	TÉC. COMERCIAL	45030
7	201308	103	6	AGENTE	103
8	201308	201	0	COMPANHIA	11030
9	201308	201	1	DIR. NACIONAL	47001
10	201308	201	2	DIR. REGIONAL	46001
11	201308	201	3	ESCR. REGIONAL	46022
12	201308	201	4	ESCR.LOCAL	46220
13	201308	201	5	TÉC. COMERCIAL	45999
14	201308	201	6	AGENTE	201

Figura 7.5: Tabela de Interface INT_COM_STR_R

Na Figura 7.5 podemos ver a tabela de interface INT_COM_STR_R. Esta tabela tem os seguintes campos:

- MEC_ANME - indica o mês em que o registo foi inserido na tabela;
- MEC_PROD - código do agente produtor;
- MEC_SEQP - nível em que o atributo MEC_ENTP se encontra na hierarquia da estrutura comercial, ilustrada na Figura 5.5. A companhia de seguros está no nível 0 que é o

mais elevado da hierarquia e o agente produtor está no nível 6 que é o mais baixo da hierarquia. Como já foi referido, o número de níveis/atributos da estrutura comercial depende da companhia de seguros;

- MEC_DCNIV - descrição do nível;
- MEC_ENTP - código do respetivo atributo na estrutura comercial.

Como se consegue perceber na Figura 7.5, os valores dos campos MEC_SEQP (ou MEC_DCNIV) deveriam ser colunas (atributos). Assim, implementou-se um *stored procedure* que transforma os valores únicos da coluna MEC_SEQP em múltiplas colunas e carrega os dados para a tabela representada na Figura 7.6. Essa tabela tem que ser criada dinamicamente sempre que se executa este *stored procedure*, porque como já foi referido anteriormente, o número de elementos na estrutura comercial varia de companhia para companhia e, por isso, o número de colunas nesta tabela não é fixo.

	ANOMES	AGENTE	NIVEL0	NIVEL1	NIVEL2	NIVEL3	NIVEL4	NIVEL5
1	201309	103	11030	47001	46002	46023	46231	45030
2	201309	201	11030	47001	46001	46022	46220	45999

Figura 7.6: Tabela Intermédia OPE_COM_STR_R

Para transformar os valores únicos da coluna MEC_SEQP em múltiplas colunas, foi utilizado o operador PIVOT no SQL Server 2008. As descrições dos códigos dos atributos da estrutura comercial são inseridos posteriormente na tabela OPE_COM_STR_R.

No final, depois de gerada dinamicamente a tabela OPE_COM_STR_R com os dados resultantes, a tabela DIMENSAO_EST_COM, ilustrada na Figura 6.20 na Secção 6.4, é limpa e carregada diretamente a partir dessa tabela através de um *Job* no Talend.

Capítulo 8

Demonstração na Plataforma MicroStrategy

Depois da implementação de toda a estrutura do data warehouse e das métricas no MicroStrategy (no Apêndice C), passou-se ao desenvolvimento dos relatórios base. Nesta secção vamos apresentar os relatórios de estatística de apólices, de receita e de custos com sinistros desenvolvidos na plataforma MicroStrategy, e mostrar alguns exemplos de *dashboards* criados a partir destes relatórios.

A Figura 8.1 apresenta o relatório de receita. Este relatório mostra os dados por ano e mês contabilístico, do valor da receita em prémios.

As métricas são: prémios brutos, prémios totais cobrados e em cobrança (líquidos de estornos). Esta receita está ainda subdividida por ramo, produto e titular (cliente).

Com este relatório extrai-se com grande facilidade (através de operações OLAP) e usando os atributos:

- ramo e produto:
 - os prémios emitidos, cobrados e em cobrança por mês;
- ramo, produto e titular:
 - os principais clientes, ordenando-os por prémio bruto emitido (dimensão da carteira do cliente);
 - os principais devedores, ordenando-os pela métrica prémio total em cobrança*.

Receita (prêmios brutos emitidos, cobrados, em cobrança)

Arquivo Editar Exibir Inserir Formato Dados Planilha Mover Janela Ajuda

Figura 8.1: Relatório de receita

Com este relatório poder-se-á construir ainda diversos *dashboards*, nomeadamente:

- evolução mensal dos prémios brutos emitidos por ramo e produto (visão e crescimento dos diferentes ramos e produtos ao longo dos meses).
- número de clientes titulares com carteira constituída por um, dois ou mais ramos/produ-
tos**.

Resumindo, um simples relatório de receita poderá ser tão útil a uma direcção financeira (área de cobrança)* como a uma direcção comercial (ação de *cross-selling*)**.

A Figura 8.2 apresenta um gráfico de evolução de prémios brutos emitidos, que ilustra o contributo dos diferentes ramos ao longo dos meses, com variações distintas, permitindo, por exemplo, verificar se há sazonalidade de vendas em algum ramo ou produto.

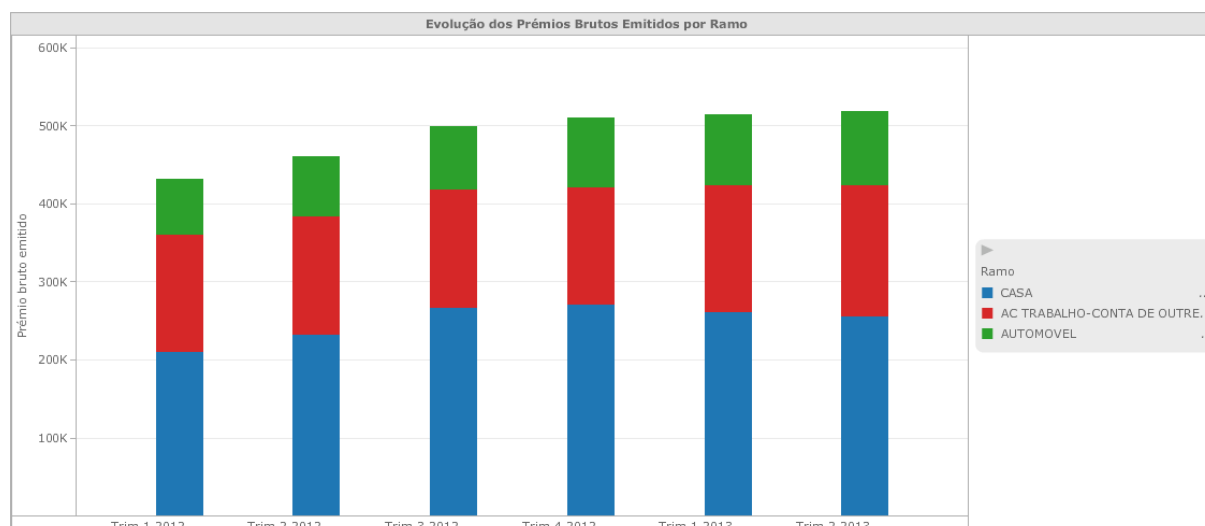


Figura 8.2: Gráfico de receita

A Figura 8.3 apresenta o relatório de estatística de apólices. Este relatório usa as métricas que indicam a quantidade de apólices registradas, novas, em vigor e anuladas e, por isso, julgamos que poderá ser útil a uma direção comercial. Com os atributos "mês do ano", "ramo", "produto", "titular" e "agente produtor" pode-se retirar estatística por agente, ramo e produto. Pela métrica "quantidade de apólices novas", os utilizadores poderão ordenar as vendas do ano em curso por:

- ramo e produto,
- agente produtor

para aferir os ramos/produtos e agentes com maior sucesso de vendas. Pela métrica "quantidade de apólices em vigor", os utilizadores conseguirão, usando o atributo "agente produtor", analisar o crescimento ou a redução das carteiras dos diferentes agentes, tanto em números absolutos como indo ao detalhe do ramo e produto.

Usando os atributos "produto" e "titular", a métrica "quantidade de apólices em vigor", é possível, com o auxílio de um gráfico, obter a taxa de penetração nos clientes da carteira de uma companhia de seguros ou de um agente produtor. Ao falarmos em taxa de penetração referimo-nos a analisar quantidades de clientes monoproduto, com duas apólices ou mais (ou taxa de penetração por ramo).

A competitividade dos diferentes produtos da seguradora poderá ser analisada nas métricas "quantidades de apólices novas" e "quantidades de apólices anuladas". Com o atributo "agente produtor" e as várias métricas deste relatório permitirão concluir sobre comportamentos e dinâmicas de vendas, como por exemplo, agentes que vendem todos os ramos versus outros que apenas vendem automóvel. Vendas de apólices novas, apólices em vigor e anulações por agente permitirão a um gestor tomar decisões atempadamente.

Estatística de apólices										
Arquivo Editar Exibir Inserir Formato Dados Planilha Mover Janela Ajuda										
Filtro de exibição										
Ano: 2012										
Mês do Ano	Ramo	Produto	Titular	Agente Produtor	Métricas	Quantidade de apólices registadas	Quantidade de apólices novas	Quantidade de apólices em vigor	Quantidade de apólices anuladas	Quantidade de apólices repostas em vigor
Janeiro	AUTOMOVEL	AUTOMOVEL	CTE CONSORCIO TECNICO ELEITORAL	EMILIO LOUREIRO E CA LDA		0	0	1	0	0
		I2S AUTO	ALEXANDRE PEREIRA	JULIO GONÇALO FERREIRA		0	0	1	0	0
			JORGE JESUS	JOÃO CARLOS FONSECA		0	0	1	0	0
Fevereiro	AUTOMOVEL	AUTOMOVEL	CTE CONSORCIO TECNICO ELEITORAL	EMILIO LOUREIRO E CA LDA		0	0	1	0	0
		I2S AUTO	ALEXANDRE PEREIRA	JULIO GONÇALO FERREIRA		0	0	1	0	0
			JORGE JESUS	JOÃO CARLOS FONSECA		0	0	1	0	0
Março	AC TRABALHO-CONTA DE OUTREM	AC.TRABALHO-C.OUTRÉM-PR.FIXO	INVESTIMENTOS EMPRESAS	JOSÉ SEGURO		2	2	0	0	0
			CHUVADE PROMOÇÕES LDA	JOSÉ ANTUNES		2	1	0	0	0
	DOENÇA-INDIVIDUAL	DOENÇA - INDIVIDUAL	GUILHERMINA RAMOS	JOSÉ SEGURO		1	1	0	0	0
			INVESTIMENTOS EMPRESAS	JORGE FREITAS LEMOS		1	1	0	0	0
	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL	INVESTIMENTOS EMPRESAS	JOSÉ SEGURO		1	1	0	0	0
			CHAUVADE PROMOÇÕES LDA	JOSÉ SEGURO		1	1	0	0	0
	MERC TRANSPORTADAS/MARITIMO	51 - MERC TRANSPORT/MARITIMO	INVESTIMENTOS EMPRESAS	JOSÉ SEGURO		1	1	0	0	0
			ALEXANDRE VERISSIMO DA ROCHA	JOSÉ SEGURO		1	1	0	0	0
	MERC TRANSPORTADAS/TERRESTRE	MULTI-PROTECÇÃO AUTO	INVESTIMENTOS EMPRESAS	JOSÉ SEGURO		1	1	0	0	0
			JORGE FREITAS LEMOS	JOSÉ ANTUNES		1	1	0	0	0
	CRÉDITOS DOC. INTERNACIONAIS	TRANSPORTE DE MERCADORIAS	CHAUVADE PROMOÇÕES LDA	JOSÉ ANTUNES		2	2	0	0	0
			CTE CONSORCIO TECNICO ELEITORAL	EMILIO LOUREIRO E CA LDA		0	0	1	0	0
Abril	AUTOMOVEL	I2S AUTO	ALEXANDRE PEREIRA	JULIO GONÇALO FERREIRA		0	0	1	0	0
			JORGE JESUS	JOÃO CARLOS FONSECA		0	0	1	0	0
	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL	JOSE MOURINHOS	PABLO MOREIRA FONSECA		2	2	0	0	0
			JOSE MOURINHO	PABLO MOREIRA FONSECA		1	1	0	0	0
	MERC TRANSPORTADAS/MARITIMO	51 - MERC TRANSPORT/MARITIMO	INVESTIMENTOS EMPRESAS	PABLO MOREIRA FONSECA		2	2	0	0	0
			MARIA JOANA PINTO	JOSÉ ANTUNES		3	3	0	0	0
	AUTOMOVEL	I2S CASA	CTE CONSORCIO TECNICO ELEITORAL	EMILIO LOUREIRO E CA LDA		0	0	2	0	0
			ANA SILVA	JOSÉ ANTUNES		4	4	0	0	0
	AUTOMOVEL	I2S AUTO	DOS SANTOS PIRES PAULO CARMELINA	PABLO MOREIRA FONSECA		1	1	0	0	0
			ALEXANDRE PEREIRA	JULIO GONÇALO FERREIRA		0	0	2	0	0
	AUTOMOVEL	I2S AUTO	JORGE JESUS	JOÃO CARLOS FONSECA		0	0	2	0	0
			CTE CONSORCIO TECNICO ELEITORAL	EMILIO LOUREIRO E CA LDA		0	0	2	0	0
Maio	AUTOMOVEL	I2S AUTO	C.E.E.A - CENTRO DE ESTUDOS ESTRATÉGICOS	JORGE FREITAS LEMOS		0	0	0	1	0
			ATLAS VIAGENS LDA	JORGE FREITAS LEMOS		0	0	0	1	0
			ALEXANDRE PEREIRA	JULIO GONÇALO FERREIRA		0	0	2	0	0
Junho	AUTOMOVEL	I2S AUTO	JORGE JESUS	JOÃO CARLOS FONSECA		0	0	2	0	0
			CTE CONSORCIO TECNICO ELEITORAL	EMILIO LOUREIRO E CA LDA		0	0	3	0	0
			ALEXANDRE PEREIRA	JULIO GONÇALO FERREIRA		0	0	3	0	0
Julho	AUTOMOVEL	I2S AUTO	JORGE JESUS	JOÃO CARLOS FONSECA		0	0	3	0	0
			CTE CONSORCIO TECNICO ELEITORAL	EMILIO LOUREIRO E CA LDA		0	0	2	0	0
			ALEXANDRE PEREIRA	JULIO GONÇALO FERREIRA		0	0	2	0	0
			JORGE JESUS	JOÃO CARLOS FONSECA		0	0	2	0	0
			CTE CONSORCIO TECNICO ELEITORAL	JORGE FREITAS LEMOS		6	6	0	0	0
			ATLAS VIAGENS LDA	JORGE FREITAS LEMOS		2	2	0	0	0

Figura 8.3: Relatório de estatística de apólices

Na Figura 8.4 temos um *dashboard* desenvolvido a partir do relatório de estatística de apólice. Neste *dashboard*, o gráfico superior ilustra a evolução do número de apólices em vigor (saldo), por ramo ao longo dos meses. O gráfico inferior apresenta a venda de apólices novas e respetivas anulações mensais.

A Figura 8.5 apresenta o relatório de custos com sinistros. Para apuramento destes custos necessitamos das seguintes métricas: variação da provisão no período, indemnizações paga e reembolsos recebidos. À soma da variação da provisão com indemnizações pagas se lhe subtrairmos os reembolsos recebidos obtemos o custo com sinistros no período. Analisando as diferentes métricas poderemos extrair a evolução mensal das indemnizações pagas, das variações das provisões ou o saldo da provisão (avaliar a velocidade de regularização dos sinistros e/ou picos de sinistralidade).

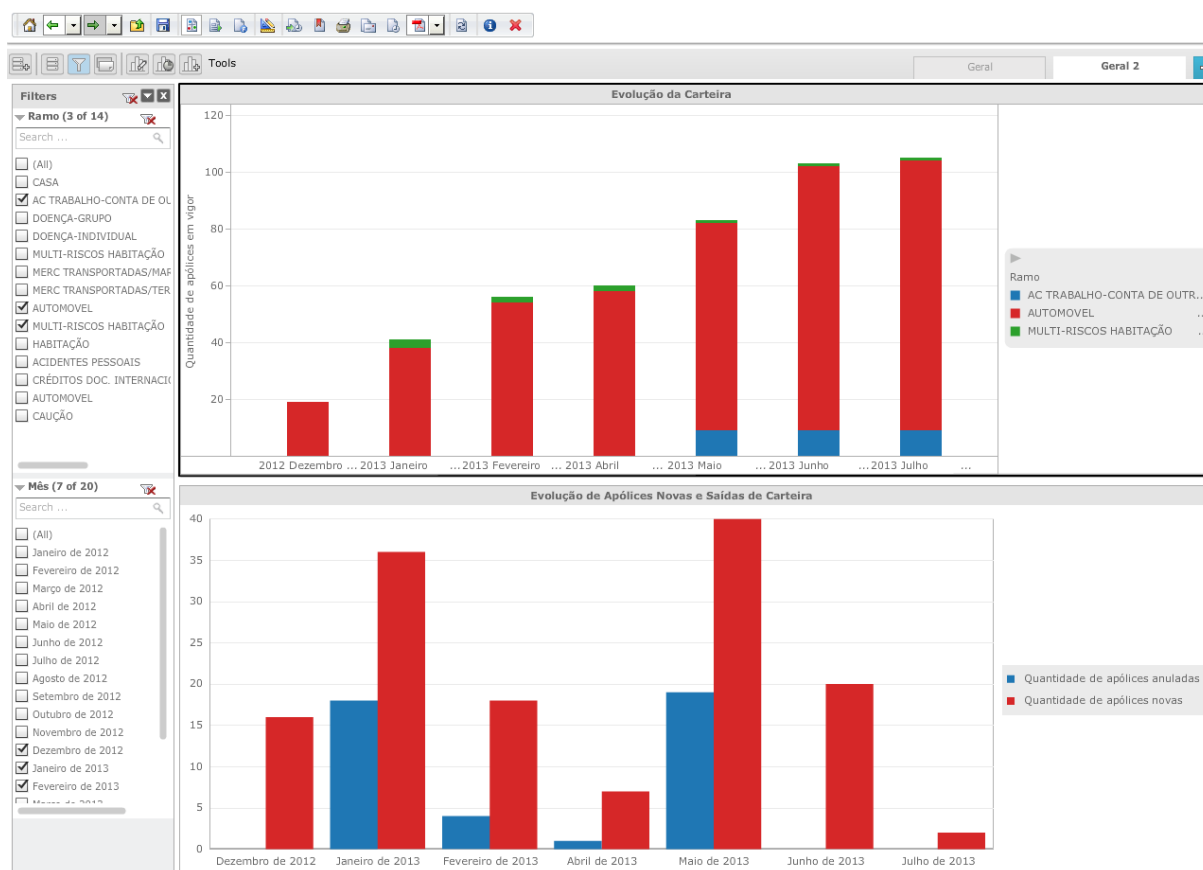


Figura 8.4: Dashboard de estatística apolice

Custos com sinistros

Arquivo Editar Exibir Inserir Formatar Dados Planilha Mover Janela Ajuda

Salvar e Fechar

Objetos do relatório

Nome	Tipo
Ano	Atributo
Mês	Atributo
Mês do Ano	Atributo
Produto	Atributo
Ramo	Atributo
Trimestre	Atributo
Custo do exercício	Métrica
Indemnizações pagas	Métrica
Reembolsos recebidos	Métrica
Saldo da provisão	Métrica
Variação da provisão no período	Métrica

Navegador de Objetos

Explorador de dados - Projeto

Nome	Tipo
Apólice	Hierarquia
Estrutura Comercial	Hierarquia
Geografia	Hierarquia
Indemnização	Hierarquia
Ramo Contabilístico até Cobertura	Hierarquia
Sinistro	Hierarquia
Tempo	Hierarquia
Titular	Hierarquia
Hierarquia de Sistema	Hierarquia do Sistema

Detalhes do Relatório

Filtro de exibição

Ano: 2012

Mês do Ano	Ramo	Produto	Métricas	Variação da provisão no período	Indemnizações pagas	Reembolsos recebidos	Saldo da provisão	Custo do exercício
Março	CASA	AT FIXO		241.600,00 €	0,00 €	0,00 €	0,00 €	241.600,00 €
	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		123.000,00 €	0,00 €	0,00 €	0,00 €	123.000,00 €
	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL		6.620,00 €	248,00 €	0,00 €	0,00 €	6.868,00 €
	AUTOMOVEL	Q2S AUTO		1.234.727,00 €	610.217,00 €	326.144,00 €	0,00 €	1.518.800,00 €
	MULTI-RISCOS HABITAÇÃO	Q2S CASA		52.640,01 €	280,00 €	0,00 €	0,00 €	52.920,01 €
	ACIDENTES PESSOAIS	ACIDENTES PESSOAIS		40.100,00 €	0,00 €	0,00 €	0,00 €	40.100,00 €
Setembro	CASA	AT FIXO		1.063.892,11 €	211.480,00 €	151.273,00 €	0,00 €	1.124.099,11 €
	AC TRABALHO-CONTA DE OUTREM	AT VARIÁVEL		372.845,19 €	9.046,87 €	0,00 €	0,00 €	381.892,06 €
	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		174.340,95 €	54.146,00 €	101.823,00 €	0,00 €	126.663,95 €
	AUTOMOVEL	Q2S AUTO		500,00 €	190,00 €	0,00 €	0,00 €	690,00 €
Novembro	MULTI-RISCOS HABITAÇÃO	Q2S CASA		0,00 €	580,00 €	0,00 €	0,00 €	580,00 €
	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		40.912,00 €	300,00 €	0,00 €	0,00 €	41.212,00 €
Dezembro	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		7.000,00 €	0,00 €	0,00 €	0,00 €	7.000,00 €
Total				3.358.177,26 €	886.487,87 €	579.240,00 €	0,00 €	3.665.425,13 €

Figura 8.5: Relatório de custos com sinistros

A sinistralidade de alguns ramos, frequência e taxa, isto é, o número de sinistros e o volume de indemnizações reclamados apresentam frequentemente picos de sinistralidade em determinados períodos do ano. Um inverno rigoroso agravará necessariamente a sinistralidade dos ramos automóvel e multiriscos de habitação. Assim, qualquer estudo sobre sinistralidade deve comparar períodos homólogos.

Custos com sinistros, indemnizações pagas, variações e saldos de provisões devem portanto ser sempre comparados em períodos homólogos. No *dashboard* ilustrado na Figura 8.6, os gráficos de barras ilustram a evolução dos custos de um mesmo mês em dois anos consecutivos. O gráfico circular detalha estes custos por ramo.

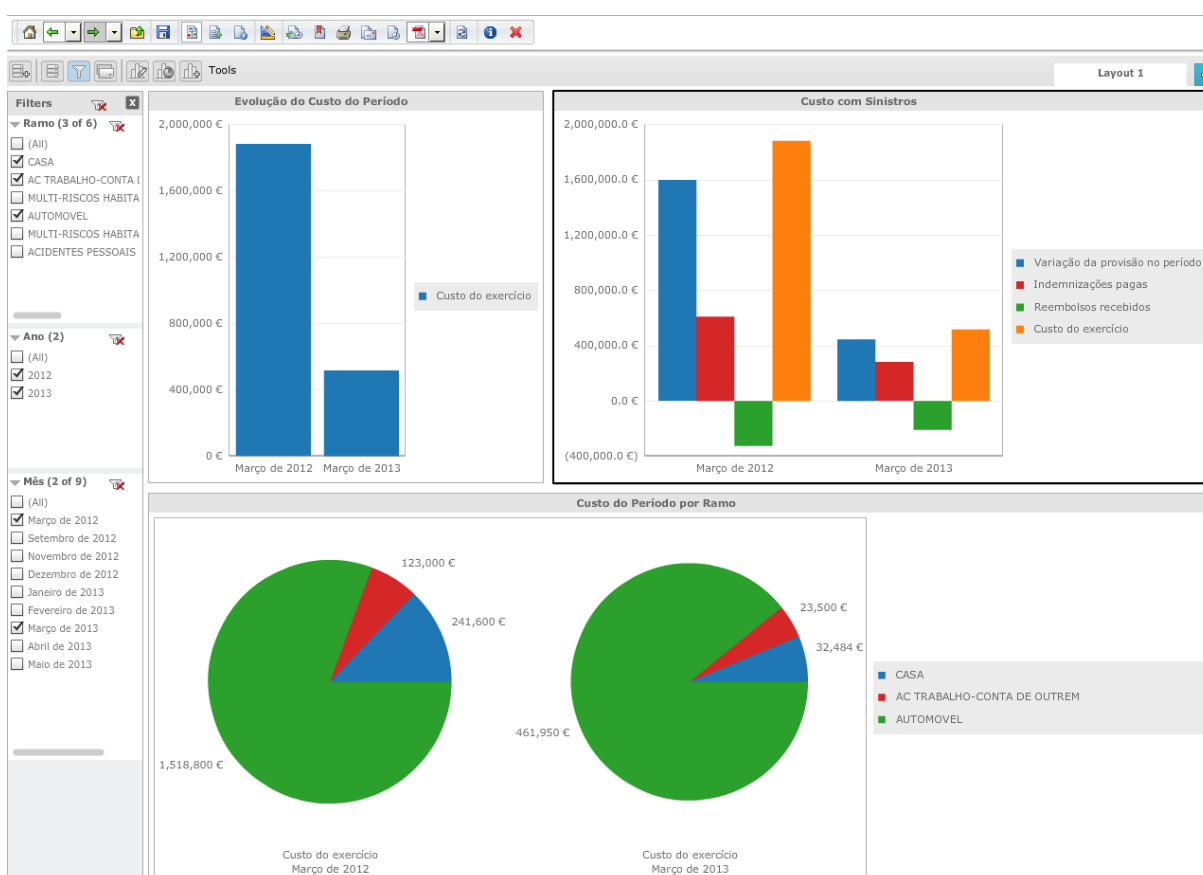


Figura 8.6: *Dashboard* de sinistros

Os relatórios, gráficos e *dashboards* acima ilustrados foram retirados de um ambiente de desenvolvimento e testes, com tabelas com pouca informação e dados pouco consistentes.

Se tivéssemos tido acesso a um ambiente de produção, nomeadamente aos dados de uma companhia de seguros ou de um corretor de seguros, poderíamos ter apresentado relatórios mais detalhados e conclusivos, podendo conter informação entre relatórios de muitos processos de negócio.

Ter-se-ia cruzado relatórios de:

- vendas (número de apólices) com prêmios, distribuídos por ramos/produtos/agentes, para uma avaliação detalhada da evolução dos respectivos prêmios médios;
- número de apólices com número de sinistros, distribuídos por ramos/produtos/agentes, para análise da frequência de sinistralidade (número sinistros/número de apólices) das diferentes métricas;

Capítulo 9

Conclusão e Trabalho Futuro

9.1 Conclusão

Neste relatório apresentámos o trabalho elaborado no âmbito de um estágio de mestrado realizado na empresa i2S - Insurance Software Solutions , e que decorreu de 14 de janeiro a 27 de setembro de 2013. Descrevemos a solução de Business Intelligence (BI) para seguros desenvolvida e que esperamos poder corresponder às expectativas dos clientes da empresa i2S, que necessitam de um sistema robusto de BI para melhorarem os seus processos de análise e consequentemente otimizarem as suas decisões.

Neste projeto foi concebido um data warehouse (DW) para o setor de negócio de seguros não vida, que irá permitir que os clientes da empresa I2S possam efetuar diversas análises ao longo de vários processos de negócio, nomeadamente análise da carteira, prémios e sinistros. Conforme explicámos na Secção 4.4, ainda que a sua arquitetura seja inspirada no modelo de Kimball, restrições impostas pela situação atual da empresa impossibilitaram uma adoção exata dessa arquitetura.

Foi também desenvolvido um processo de ETL (Capítulo 7) para extrair dados de uma base de dados intermédia e os carregar no DW, realizando algumas transformações. Para a sua implementação, recorreu-se maioritariamente à ferramenta de integração de dados Talend.

Por fim, usando a plataforma de BI da MicroStrategy, foram definidos um conjunto de métricas, relatórios e *dashboards*, que efetuam a consulta do DW desenvolvido.

Como referimos na Secção 4.1, com este projeto, os clientes da I2S vão poder adquirir um produto que se pode classificar na etapa 4 do modelo de maturidade proposto pelo instituto TDWI, o que constitui um melhoramento da situação atual.

9.2 Trabalho Futuro

O trabalho desenvolvido, apesar de consistir num sistema de BI útil e funcional que se adapta às necessidades de uma companhia de seguros, encontra-se em constante evolução. Este projeto só entra no mercado em janeiro de 2014 com a sua instalação numa seguradora, e, por isso, ainda há tempo para implementar algumas destas possíveis melhorias:

- apesar de já terem sido feitos alguns testes ao sistema, é necessário uma validação mais detalhada para entrada em produção;
- procurar formas de otimizar o sistema, realizando vários testes de desempenho;
- exploração dos métodos de data mining no MicroStrategy, para gerar mais informação útil, de forma a que as companhias de seguros consigam tomar melhores decisões de negócio;
- melhorar a arquitetura definida, implementando um processo que aceda diretamente aos sistemas operacionais, sem passar pela base de dados Informacional;
- inserir as dimensões moeda e taxa de câmbio, porque em alguns mercados (como Angola e Moçambique) circula mais do que uma moeda;
- integrar mais processos de negócio do setor não vida no data warehouse, como resseguro, ou de outros setores como o vida ou a área administrativa financeira.

Glossário

Atributo - Coluna numa tabela de dimensão [20].

Business Intelligence - "Termo genérico que inclui aplicações, infraestruturas, ferramentas e boas práticas que permitem o acesso e a análise de informações para melhorar e otimizar decisões e desempenho" [15].

Chaves Substitutivas (*Surrogate Keys*) - Chaves geradas no processo de ETL para popular as tabelas de dimensão. Normalmente estas chaves são números inteiros sequenciais e não têm significado de negócio. São utilizadas como chaves primárias nas tabelas de dimensão e como chaves secundárias na tabela de factos [20].

Data Mart - Subconjunto lógico de um data warehouse que contém um conjunto de dados relativos a uma atividade ou a um processo de negócio específico.

Data Warehouse - Integração de dados consolidados provenientes de múltiplas e possivelmente heterogéneas fontes de dados para o propósito de apoio à tomada de decisão [29].

Data Warehousing - Processo de construir e gerir data warehouses.

Dimensão - "Entidade independente num modelo dimensional que serve como ponto de entrada ou como mecanismo de operações de *slice and dice* das medidas aditivas localizadas na tabela de factos do modelo dimensional" [20].

Esquema em Floco de Neve (*Snowflake Schema*) - Variação do esquema em estrela, onde as tabelas de dimensão são normalizadas, resultando em múltiplas tabelas relacionadas entre si.

Esquema em Estrela (*Star Schema*) - "Representação genérica de um modelo dimensional numa base de dados relacional, em que uma tabela de factos com uma chave composta é ligada a um número de tabelas de dimensão, cada uma com uma chave primária única" [20].

Facto - "Medida de desempenho do negócio, tipicamente numérica e aditiva, que está armazenada numa tabela de factos" [20].

Filtro - Objeto que serve para especificar ou limitar os dados apresentados, por exemplo, num relatório, utilizando condições com operadores lógicos.

Hierarquia - "Árvore direcionada, enraizada numa dimensão, cujos nós são todos os atributos que descrevem essa dimensão, e onde os ramos modelam associações de muitos-para-um entre os pares de atributos" [29].

Métrica - Cálculo analítico de negócio, normalmente implementado utilizando factos, em que o seu resultado é apresentado nos relatórios.

MicroStrategy - Plataforma de Business Intelligence da empresa MicroStrategy.

Modelo Dimensional - Técnica de desenho lógico muitas vezes utilizada em *data warehouses*. O modelo dimensional não envolve necessariamente uma base de dados relacional. A modelação dimensional utiliza sempre os conceitos de factos e dimensões [19].

Processo ETL (*Extract, Transform and Load*) - Processo de data warehousing que consiste na extração (i.e., ler os dados de uma ou mais bases de dados), transformação (i.e., converter os dados extraídos de modo a que possam ser colocados num data warehouse ou, simplesmente, numa outra base de dados) e carregamento (i.e., armazenar os dados num data warehouse) [27].

Prompt - Objeto utilizado para questionar um utilizador durante a execução de um relatório. A resposta dada pelo utilizador vai determinar que dados são apresentados no relatório.

Sistemas OLTP (*Online Transaction Processing*) - "Sistema de transação que é o principal responsável pela captura e armazenamento de dados relacionados com as funções de negócio do dia a dia" [27].

Tabela de Dimensão - Tabela de um modelo dimensional com uma única chave primária e atributos descritivos [20].

Tabela de Factos - Tabela central num esquema em estrela ou num esquema em floco de neve (modelo dimensional) com medidas numéricas de desempenho caracterizadas por uma chave composta, em que cada um dos seus elementos é uma chave secundária elaborada a partir de uma tabela de dimensão [20].

Trigger (base de dados) - Procedimento armazenado que é automaticamente executado quando ocorrem certos eventos numa tabela ou view de uma base de dados.

View (SQL) - Instrução SQL armazenada na base de dados. É uma representação virtual de uma tabela composta por um conjunto de resultados retornados por uma consulta a uma ou mais tabelas reais.

Apêndice A

Acrónimos

3FN - Terceira Forma Normal

BI - Business Intelligence

BD - Base de Dados

DD - Data Discovery

DM - Data Mart

DW - Data Warehouse

DWB - Data Warehouse Bus

ETL - Extract, Transform e Load

KPI - Key Performance Indicator

MSTR - MicroStrategy

OLAP - Online Analytical Processing

ROLAP - Relational Online Analytical Processing

OLTP - Online Transaction Processing

SQL - Structured Query Language

Apêndice B

Tabelas de Factos de Inventário

As Figuras B.1 - B.4 representam quatro esquemas em estrela que compõem um subconjunto do modelo dimensional desenvolvido. Nestas figuras também é possível observar os campos das tabelas de factos desses esquemas.

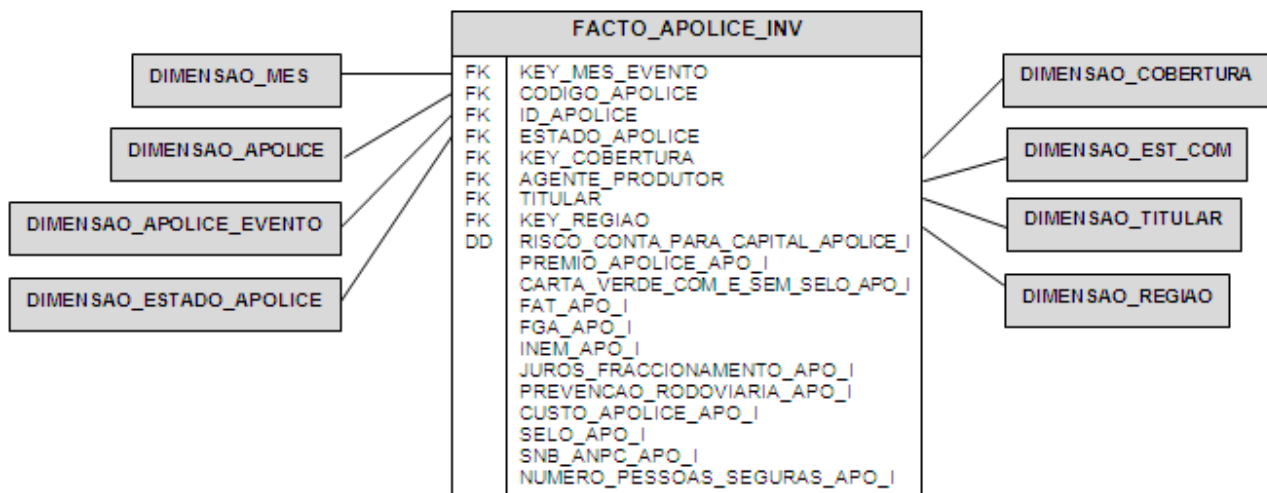


Figura B.1: Tabela de Factos de Apólice de Inventário

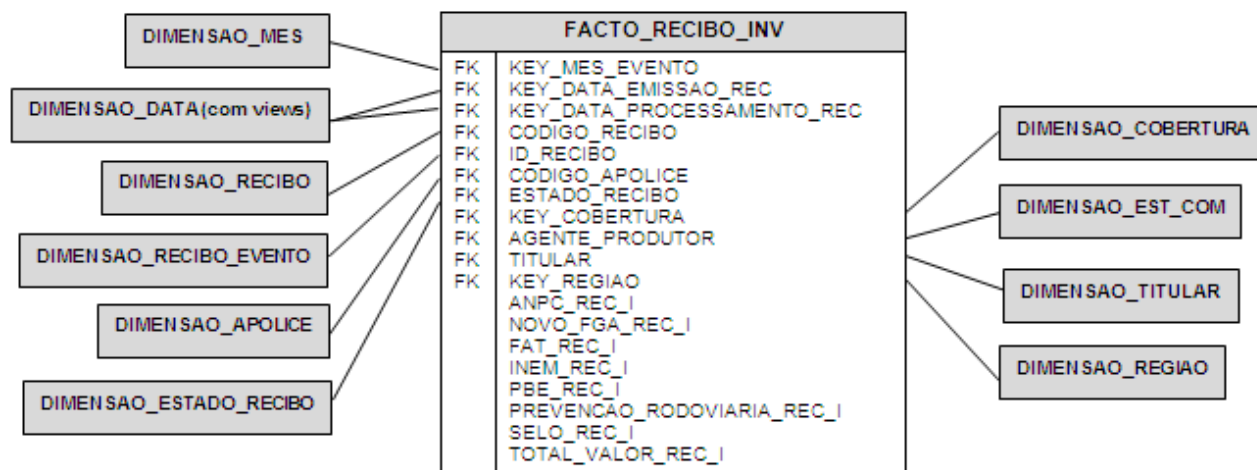


Figura B.2: Tabela de Factos de Recibo de Inventário



Figura B.3: Tabela de Factos de Sinistro de Inventário

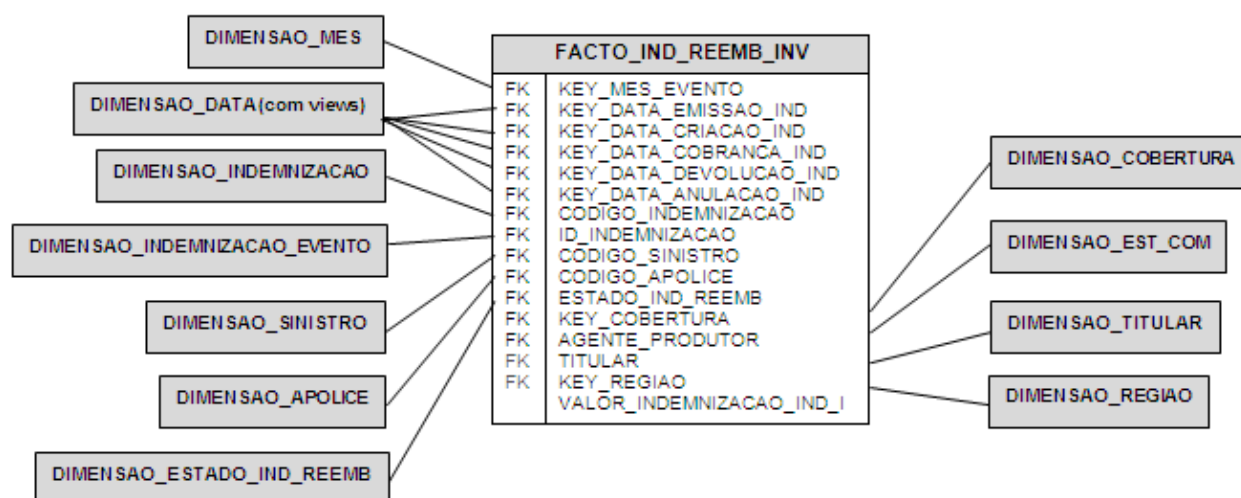


Figura B.4: Tabela de Factos de Indemnização/Reembolso de Inventário

Apêndice C

Métricas e Relatórios Desenvolvidos no MSTR

Métricas desenvolvidas no MicroStrategy:

Nome /
Pessoas seguras - apólices anuladas
Pessoas seguras - apólices em vigor
Pessoas seguras - apólices novas
Pessoas seguras - apólices potenciais
Pessoas seguras - apólices registadas
Pessoas seguras - apólices repostas em vigor
Prémio - apólices anuladas
Prémio - apólices em vigor
Prémio - apólices novas
Prémio - apólices potenciais
Prémio - apólices registadas
Prémio - apólices repostas em vigor
Prémio bruto anual - apólices anuladas
Prémio bruto anual - apólices em vigor
Prémio bruto anual - apólices novas
Prémio bruto anual - apólices potenciais
Prémio bruto anual - apólices registadas
Prémio bruto anual - apólices repostas em vigor
Prémio total anual - apólices anuladas
Prémio total anual - apólices em vigor
Prémio total anual - apólices novas
Prémio total anual - apólices potenciais
Prémio total anual - apólices registadas
Prémio total anual - apólices repostas em vigor
Quantidade de apólices anuladas
Quantidade de apólices anuladas por cobertura
Quantidade de apólices diferidas
Quantidade de apólices diferidas por cobertura
Quantidade de apólices em vigor
Quantidade de apólices em vigor por cobertura
Quantidade de apólices novas
Quantidade de apólices novas por cobertura
Quantidade de apólices potenciais
Quantidade de apólices potenciais por cobertura
Quantidade de apólices registadas
Quantidade de apólices registadas por cobertura
Quantidade de apólices repostas em vigor
Quantidade de apólices repostas em vigor por cobertura

Figura C.1: Métricas de Apólice

Nome /
Custo sinistros
Custo sinistros pendentes
Desvios negativos de provisão
Desvios positivos de provisão
Indemnizações anuladas
Indemnizações em pagamento
Indemnizações pagas
Indemnizações pagas líquidos de reembolsos
Indemnizações processadas
Pensões em curso
Pensões pagas
Provisão para indemnizações
Provisão para reembolsos
Provisões matemáticas
Quantidade de sinistros abertos
Quantidade de sinistros abertos por cobertura
Quantidade de sinistros anulados
Quantidade de sinistros anulados por cobertura
Quantidade de sinistros em contencioso
Quantidade de sinistros em curso administrativamente
Quantidade de sinistros em curso administrativamente por cobertura
Quantidade de sinistros em curso tecnicamente
Quantidade de sinistros em curso tecnicamente por cobertura
Quantidade de sinistros encerrados
Quantidade de sinistros encerrados por cobertura
Quantidade de sinistros pagos (Indemnizações)
Quantidade de sinistros pagos (Reembolsos)
Quantidade de sinistros reabertos
Quantidade de sinistros reabertos por cobertura
Quantidade de sinistros reencerrados
Quantidade de sinistros reencerrados por cobertura
Rácio de sinistralidade
Reembolsos anulados
Reembolsos em recebimento
Reembolsos processados
Reembolsos recebidos
Saldo da provisão
Saldo da provisão no período
Saldo da provisão para pagamentos
Saldo da provisão para pagamentos no período
Saldo da provisão para reembolsos
Saldo da provisão para reembolsos no período
Variação da provisão no período

Figura C.2: Métricas de Sinistros

Nome /	Tipo
ANPC - estornos anulados	Métrica
ANPC - estornos cobrados	Métrica
ANPC - estornos em pagamento	Métrica
ANPC - estornos processados	Métrica
ANPC - recibos anulados	Métrica
ANPC - recibos cobrados	Métrica
ANPC - recibos em cobrança	Métrica
ANPC - recibos processados	Métrica
ANPC cobrado líquido de estornos	Métrica
ANPC líquido de estornos e anulações	Métrica
FAT - estornos anulados	Métrica
FAT - estornos cobrados	Métrica
FAT - estornos em pagamento	Métrica
FAT - estornos processados	Métrica
FAT - recibos anulados	Métrica
FAT - recibos cobrados	Métrica
FAT - recibos em cobrança	Métrica
FAT - recibos processados	Métrica
FAT cobrado líquido de estornos	Métrica
FAT líquido de estornos e anulações	Métrica
FGA - estornos anulados	Métrica
FGA - estornos cobrados	Métrica
FGA - estornos em pagamento	Métrica
FGA - estornos processados	Métrica
FGA - recibos anulados	Métrica
FGA - recibos cobrados	Métrica
FGA - recibos em cobrança	Métrica
FGA - recibos processados	Métrica
FGA cobrado líquido de estornos	Métrica
FGA líquido de estornos e anulações	Métrica
Impostos - estornos anulados	Métrica
Impostos - estornos em pagamento	Métrica
Impostos - estornos processados	Métrica
Impostos - recibos cobrados	Métrica
Impostos - recibos processados	Métrica
Impostos cobrados líquidos de estornos	Métrica
Impostos em cobrança líquido de estornos	Métrica
Impostos- estornos cobrados	Métrica
Impostos líquidos de estornos e anulações	Métrica
Impostos- recibos anulados	Métrica
Impostos- recibos em cobrança	Métrica

Figura C.3: Métricas de Impostos

INEM - estornos anulados	Métrica
INEM - estornos cobrados	Métrica
INEM - estornos em pagamento	Métrica
INEM - estornos processados	Métrica
INEM - recibos anulados	Métrica
INEM - recibos cobrados	Métrica
INEM - recibos em cobrança	Métrica
INEM - recibos processados	Métrica
INEM cobrado líquido de estornos	Métrica
INEM líquido de estornos e anulações	Métrica
Prevenção rodoviária - estornos anulados	Métrica
Prevenção rodoviária - estornos cobrados	Métrica
Prevenção rodoviária - estornos em pagamento	Métrica
Prevenção rodoviária - estornos processados	Métrica
Prevenção rodoviária - recibos anulados	Métrica
Prevenção rodoviária - recibos cobrados	Métrica
Prevenção rodoviária - recibos em cobrança	Métrica
Prevenção rodoviária - recibos processados	Métrica
Prevenção rodoviária cobrado líq. de estornos	Métrica
Prevenção rodoviária líq. de estornos e anulações	Métrica
Selo de apólice - estornos anulados	Métrica
Selo de apólice - estornos cobrados	Métrica
Selo de apólice - estornos em pagamento	Métrica
Selo de apólice - estornos processados	Métrica
Selo de apólice - recibos anulados	Métrica
Selo de apólice - recibos cobrados	Métrica
Selo de apólice - recibos em cobrança	Métrica
Selo de apólice - recibos processados	Métrica
Selo de apólice cobrado líquido de estornos	Métrica
Selo de apólice líquido de estornos e anulações	Métrica

Figura C.4: Métricas de Impostos

Nome /
Apólices e actas adicionais - estornos anulados
Apólices e actas adicionais - estornos cobrados
Apólices e actas adicionais - estornos processados
Apólices e actas adicionais - recibos anulados
Apólices e actas adicionais - recibos cobrados
Apólices e actas adicionais - recibos processados
Apólices e actas adicionais líquido de estornos e anulações
Carta verde - estornos anulados
Carta verde - estornos cobrados
Carta verde - estornos processados
Carta verde - recibos anulados
Carta verde - recibos cobrados
Carta verde - recibos processados
Carta verde com selo - estornos anulados
Carta verde com selo - estornos cobrados
Carta verde com selo - estornos processados
Carta verde com selo - recibos anulados
Carta verde com selo - recibos cobrados
Carta verde com selo - recibos processados
Carta verde com selo líquido de estornos e anulações
Carta verde líquida de estornos e anulações
Carta verde sem selo - estornos anulados
Carta verde sem selo - estornos cobrados
Carta verde sem selo - estornos processados
Carta verde sem selo - recibos anulados
Carta verde sem selo - recibos cobrados
Carta verde sem selo - recibos processados
Carta verde sem selo líquida de estornos e anulações
Juros de fraccionamento - estornos anulados
Juros de fraccionamento - estornos cobrados
Juros de fraccionamento - estornos processados
Juros de fraccionamento - recibos anulados
Juros de fraccionamento - recibos cobrados
Juros de fraccionamento - recibos processados
Juros de fraccionamento COS - estornos anulados
Juros de fraccionamento COS - estornos cobrados
Juros de fraccionamento COS - estornos processados
Juros de fraccionamento COS - recibos anulados
Juros de fraccionamento COS - recibos cobrados
Juros de fraccionamento COS - recibos processados
Juros de fraccionamento COS líq.de est.e anulações
Juros de fraccionamento líq.COS - estornos anulados
Juros de fraccionamento líq.COS - estornos cobrados
Juros de fraccionamento líq.COS - estornos processados
Juros de fraccionamento líq.COS - recibos anulados
Juros de fraccionamento líq.COS - recibos cobrados
Juros de fraccionamento líq.COS - recibos processados
Juros de fraccionamento líq.COS e est.e anulações
Juros de fraccionamento líq.de est. e anulações
Prémio bruto - estornos anulados

Figura C.5: Métricas de Produção

Nome /
Prémio bruto - estornos cobrados
Prémio bruto - estornos processados
Prémio bruto - recibos anulados
Prémio bruto - recibos cobrados
Prémio bruto - recibos processados
Prémio bruto cobrado líquido de estornos
Prémio bruto em cobrança
Prémio bruto em cobrança líquido de estorno
Prémio bruto em pagamento
Prémio bruto emitido
Prémio comercial - estornos anulados
Prémio comercial - estornos cobrados
Prémio comercial - estornos processados
Prémio comercial - recibos anulados
Prémio comercial - recibos cobrados
Prémio comercial - recibos processados
Prémio comercial COS - estornos anulados
Prémio comercial COS - estornos cobrados
Prémio comercial COS - estornos processados
Prémio comercial COS - recibos anulados
Prémio comercial COS - recibos cobrados
Prémio comercial COS - recibos processados
Prémio comercial COS líq. de estornos e anulações
Prémio comercial líq. COS - estornos anulados
Prémio comercial líq. COS - estornos cobrados
Prémio comercial líq. COS - estornos processados
Prémio comercial líq. COS - recibos anulados
Prémio comercial líq. COS - recibos cobrados
Prémio comercial líq. COS - recibos processados
Prémio comercial líq. COS e estornos e anulações
Prémio comercial líquido de estornos e anulações
Prémio total - estornos anulados
Prémio total - estornos cobrados
Prémio total - estornos processados
Prémio total - recibos anulados
Prémio total - recibos cobrados
Prémio total - recibos processados
Prémio total cobrado líquido de estornos
Prémio total em cobrança
Prémio total em cobrança líquido de estornos
Prémio total em pagamento
Prémio total líquido de estornos e anulações
Quantidade de estornos anulados
Quantidade de estornos cobrados
Quantidade de estornos em pagamento
Quantidade de estornos processados
Quantidade de recibos anulados
Quantidade de recibos cobrados
Quantidade de recibos em cobrança
Quantidade de recibos processados

Figura C.6: Métricas de Produção

Nome /	Tipo
Comissões - estornos anulados	Métrica
Comissões - estornos cobrados	Métrica
Comissões - estornos processados	Métrica
Comissões - recibos anulados	Métrica
Comissões - recibos cobrados	Métrica
Comissões - recibos processados	Métrica
Comissões cobradas líquidas de estornos	Métrica
Comissões COS - estornos anulados	Métrica
Comissões COS - estornos cobrados	Métrica
Comissões COS - estornos processados	Métrica
Comissões COS - recibos anulados	Métrica
Comissões COS - recibos cobrados	Métrica
Comissões COS - recibos processados	Métrica
Comissões COS cobradas líquidas de estornos	Métrica
Comissões COS líquida de estornos e anulações	Métrica
Comissões líquida de estornos e anulações	Métrica
Comissões líquidas COS e estornos e anulações	Métrica

Figura C.7: Métricas de Comissões

Relatórios desenvolvidos no MicroStrategy:

Vendas totais por número de apólices									
Arquivo Editar Exibir Inserir Formato Dados Planilha Mover Janela Ajuda									
Detalhes do Relatório									
Filtro de exibição									
Ano: 2012									
Mês do Ano	Ramo	Produto	Escritório Local	Técnico Comercial	Agente Produtor	Métricas	Quantidade de apólices novas	Prêmio total anual - apólices novas	
Março	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO	ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	JOSÉ SEGURO		2	13.090,00 €	
			ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	JOSÉ SEGURO		1	325,47 €	
	DOENÇA-INDIVIDUAL	DOENÇA - INDIVIDUAL	ESCRITÓRIO LOCAL LISBOA	JORGE SILVA M. COSTA	JORGE FREITAS LEMOS		1	88,01 €	
			ESCRITÓRIO LOCAL PORTO	CARLOS ALBERTO GOMES	JOSÉ ANTUNES		1	941,02 €	
	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL	ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	JOSÉ SEGURO		1	21,50 €	
	MERC TRANSPORTADAS/MARITIMO	51 - MERC TRANSPORT/MARITIMO	ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	JOSÉ SEGURO		2	54,00 €	
	MERC TRANSPORTADAS/TERRESTRE	51 - MERC TRANSPORT/TERRESTRE	ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	JOSÉ SEGURO		1	592,57 €	
	AUTOMOVEL	MULTI-PROTEÇÃO AUTO	ESCRITÓRIO LOCAL LISBOA	SILVIO RESENDE	JOSÉ SEGURO		1	81,42 €	
Abril			ESCRITÓRIO LOCAL LISBOA	JORGE SILVA M. COSTA	JORGE FREITAS LEMOS		1	688,03 €	
	CRÉDITOS DOC. INTERNACIONAIS	TRANSPORTE DE MERCADORIAS	ESCRITÓRIO LOCAL LISBOA	CARLOS ALBERTO GOMES	JOSÉ ANTUNES		2	457,92 €	
	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL	ESCRITÓRIO LOCAL PORTO	TÉCNICO COMERCIAL (DEFAULT)	PABLO MOREIRA FONSECA		3	309,09 €	
	MERC TRANSPORTADAS/MARITIMO	51 - MERC TRANSPORT/MARITIMO	ESCRITÓRIO LOCAL PORTO	TÉCNICO COMERCIAL (DEFAULT)	PABLO MOREIRA FONSECA		2	54,00 €	
	MULTI-RISCOS HABITAÇÃO	25 CASA	ESCRITÓRIO LOCAL LISBOA	CARLOS ALBERTO GOMES	JOSÉ ANTUNES		3	247,20 €	
	AUTOMOVEL	25 AUTO	ESCRITÓRIO LOCAL PORTO	TÉCNICO COMERCIAL (DEFAULT)	PABLO MOREIRA FONSECA		1	0,00 €	
			ESCRITÓRIO LOCAL LISBOA	CARLOS ALBERTO GOMES	JOSÉ ANTUNES		4	0,00 €	
			ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	MÁRIO ANDRÉ GOMES		4	3.216,00 €	
Agosto	CASA	AT FIXO	ESCRITÓRIO LOCAL LISBOA	JORGE SILVA M. COSTA	JORGE FREITAS LEMOS		13	28.106,00 €	
			ESCRITÓRIO LOCAL LISBOA	JOSE ANTONIO ALVES	MARTA RODRIGUES PINHO		5	26.714,16 €	
		AT VARIÁVEL	ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	MÁRIO ANDRÉ GOMES		2	5.733,00 €	
			ESCRITÓRIO LOCAL LISBOA	JORGE SILVA M. COSTA	JORGE FREITAS LEMOS		5	5.574,96 €	
	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO	ESCRITÓRIO LOCAL PORTO	JOSE ANTONIO ALVES	MARTA RODRIGUES PINHO		3	6.439,20 €	
			ESCRITÓRIO LOCAL LISBOA	SILVIO RESENDE	MÁRIO ANDRÉ GOMES		17	8.230,02 €	
			ESCRITÓRIO LOCAL LISBOA	JORGE SILVA M. COSTA	JORGE FREITAS LEMOS		10	7.539,33 €	
			ESCRITÓRIO LOCAL PORTO	JOSE ANTONIO ALVES	MARTA RODRIGUES PINHO		1	1.716,48 €	
	DOENÇA-GRUPO	DOENÇA - GRUPO	ESCRITÓRIO LOCAL LISBOA	CARLOS ALBERTO GOMES	JOSÉ ANTUNES		1	0,00 €	
			ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	MÁRIO ANDRÉ GOMES		16	3.220,67 €	
		M.R.H-CONF.TOTAL/EXCL	ESCRITÓRIO LOCAL LISBOA	CARLOS ALBERTO GOMES	JOSÉ ANTUNES		1	153,35 €	
			ESCRITÓRIO LOCAL PORTO	JOSE ANTONIO ALVES	MARTA RODRIGUES PINHO		4	219,73 €	
	AUTOMOVEL	25 AUTO	ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	MÁRIO ANDRÉ GOMES		25	15.776,96 €	
			ESCRITÓRIO LOCAL LISBOA	JORGE SILVA M. COSTA	JORGE FREITAS LEMOS		13	7.443,20 €	
			ESCRITÓRIO LOCAL LISBOA	CARLOS ALBERTO GOMES	JOSÉ ANTUNES		3	1.868,04 €	
			ESCRITÓRIO LOCAL LISBOA	JOSE ANTONIO ALVES	MARTA RODRIGUES PINHO		5	3.137,53 €	
Setembro	MULTI-RISCOS HABITAÇÃO	25 CASA	ESCRITÓRIO LOCAL LISBOA	JOSE ANTONIO ALVES	FERNANDA ANTUNES SILVA		1	542,44 €	
			ESCRITÓRIO LOCAL PORTO	SILVIO RESENDE	MÁRIO ANDRÉ GOMES		15	2.189,26 €	
			ESCRITÓRIO LOCAL LISBOA	JORGE SILVA M. COSTA	JORGE FREITAS LEMOS		3	1.264,02 €	
			ESCRITÓRIO LOCAL LISBOA	CARLOS ALBERTO GOMES	JOSÉ ANTUNES		4	933,60 €	
	ACIDENTES PESSOAIS	ACIDENTES PESSOAIS	ESCRITÓRIO LOCAL LISBOA	JOSE ANTONIO ALVES	MARTA RODRIGUES PINHO		3	768,22 €	
			ESCRITÓRIO LOCAL PORTO	JOSE ANTONIO ALVES	MARTA RODRIGUES PINHO		1	25,26 €	
			ESCRITÓRIO LOCAL LISBOA	SILVIO RESENDE	MÁRIO ANDRÉ GOMES		26	77.199,04 €	
			ESCRITÓRIO LOCAL LISBOA	ANTONIO PEDRO A. MOURA	JOÃO MANUEL FONSECA		1	3.210,00 €	
Outubro	CASA	AT FIXO	ESCRITÓRIO LOCAL LISBOA	ANTONIO PEDRO A. MOURA	INÉS SILVA PEREIRA		1	357,50 €	
			ESCRITÓRIO LOCAL LISBOA	JORGE SILVA M. COSTA	JORGE FREITAS LEMOS		10	51.041,40 €	

Figura C.8: Relatório de vendas totais por número de apólices

Apólices em vigor - evolução

Objetos do relatório: Nome / Tipo

- Ano: Atributo
- Mês do Ano: Atributo
- Produto: Atributo
- Ramo: Atributo
- Quantidade de apólices em vigor: Métrica

Navegador de Objetos: Explorador de dados - Projeto

Detalhes do Relatório: Filtro de exibição: Ano: 2013

Mês do Ano	Janeiro	Fevereiro	Março	Abril	Maio	Junho	Julho
Métricas	Quantidade de apólices em vigor	Quantidade de apólices em vigor	Quantidade de apólices em vigor	Quantidade de apólices em vigor	Quantidade de apólices em vigor	Quantidade de apólices em vigor	Quantidade de apólices em vigor
Ramo							
AC TRABALHO-CONTA DE OUTREM							
AUTOMOVEL							
PRODUTO							
AC TRABALHO-C. OUTRÉM-PR.FIXO							
MULTI-PROTECÇÃO AUTO	29	20	10	10	9	9	9
Q25 AUTO	9	36	25	34	10	10	10
MULTIRISCOS HABITAÇÃO	9	6	3	3	3	3	3
Q25 CASA	6	4	2	2	2	2	2
AUTOMOVEL	15	10	5	5	5	5	5
Q25 AUTO	68	76	45	54	97	117	119
Total							

Figura C.9: Relatório de apólices em vigor (evolução)

Comissões

Objetos do relatório: Nome / Tipo

- Ano: Atributo
- Produto: Atributo
- Ramo: Atributo
- Trimestre: Atributo
- Comissões cobradas líquidas de estornos: Métrica
- Comissões líquida de estornos e anulações: Métrica

Navegador de Objetos: Explorador de dados - Projeto

Detalhes do Relatório: Filtro de exibição: Ano: 2012

Trimestre	Ramo	Produto	Métricas	Comissões líquida de estornos e anulações	Comissões cobradas líquidas de estornos
Trim 1 2012	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		0.00 €	0.00 €
	DOENÇA-INDIVIDUAL	DOENÇA - INDIVIDUAL		0.00 €	0.00 €
	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL		0.00 €	0.00 €
	MERC TRANSPORTADAS/MARITIMO	S1 - MERC TRANSPORT/MARITIMO		0.00 €	0.00 €
	MERC TRANSPORTADAS/TERRESTRE	S1 - MERC TRANSPORT/TERRESTRE		0.00 €	0.00 €
	AUTOMOVEL	MULTI-PROTECÇÃO AUTO		0.00 €	0.00 €
	ACIDENTES PESSOAIS	ACIDENTES PESSOAIS		0.00 €	0.00 €
Trim 2 2012	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL		0.00 €	0.00 €
	MERC TRANSPORTADAS/MARITIMO	S1 - MERC TRANSPORT/MARITIMO		0.00 €	0.00 €
	MULTI-RISCOS HABITAÇÃO	Q25 CASA		4.95 €	0.00 €
	CASA	AT FIXO		666.91 €	723.67 €
		AT VARIÁVEL		12.690.68 €	7.926.77 €
Trim 3 2012	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		2.724.72 €	315.42 €
	DOENÇA-GRUPO	DOENÇA - GRUPO		0.00 €	0.00 €
	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL		0.00 €	0.00 €
	AUTOMOVEL	MULTI-PROTECÇÃO AUTO		0.00 €	0.00 €
	MULTI-RISCOS HABITAÇÃO	Q25 AUTO		0.00 €	0.00 €
		Q25 CASA		158.27 €	28.46 €
	CASA	AT FIXO		28.75 €	0.00 €
Trim 4 2012	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		750.74 €	451.31 €
	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL		0.00 €	0.00 €
	AUTOMOVEL	MULTI-PROTECÇÃO AUTO		0.00 €	0.00 €
		Q25 AUTO		0.00 €	0.00 €
Total				17.023.02 €	9.443.63 €

Figura C.10: Relatório de comissões

Contagem de sinistros

Objetos do relatório: Nome / Tipo

- Ano: Atributo
- Mês do Ano: Atributo
- Produto: Atributo
- Ramo: Atributo
- Quantidade de sinistros abertos: Métrica
- Quantidade de sinistros anulados: Métrica
- Quantidade de sinistros encerrados: Métrica
- Quantidade de sinistros reabertos: Métrica
- Quantidade de sinistros reencerrados: Métrica
- Quantidade de sinistros em curso: Métrica
- Quantidade de sinistros em curso tecnicamente: Métrica

Navegador de Objetos: Explorador de dados - Projeto

Detalhes do Relatório: Filtro de exibição: Ano: 2012

Mês do Ano	Ramo	Produto	Métricas	Quantidade de sinistros abertos	Quantidade de sinistros anulados	Quantidade de sinistros encerrados	Quantidade de sinistros reabertos	Quantidade de sinistros reencerrados	Quantidade de sinistros em curso	Quantidade de sinistros em curso tecnicamente
Março	CASA	AT FIXO		32						
	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		22						
	MULTI-RISCOS HABITAÇÃO	M.R.H-CONF.TOTAL/EXCL		4						
	AUTOMOVEL	Q25 AUTO		55						
	MULTI-RISCOS HABITAÇÃO	Q25 CASA		14		1				
	ACIDENTES PESSOAIS	ACIDENTES PESSOAIS		3						
Setembro	CASA	AT FIXO		73		3	2			
		AT VARIÁVEL		23	1	1				
	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		68		1				
	AUTOMOVEL	Q25 AUTO		1						
Novembro	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		13						
Dezembro	AC TRABALHO-CONTA DE OUTREM	AC TRABALHO-C. OUTRÉM-PR.FIXO		4						
Total				310	3	6	2	0	0	0

Figura C.11: Relatório de contagem de sinistros

Referências

- [1] Thilini Ariyachandra and Hugh Watson. Key organizational factors in data warehouse architecture selection. *Decision Support Systems*, 49(2):200–212, 2010.
- [2] Mary Breslin. Data Warehousing Battle of the Giants: Comparing the Basics of the Kimball and Inmon Models. *Business Intelligence Journal*, pages 6–20, Winter 2004.
- [3] Jan Buzydlowski, Il-Yeol Song, and Lewis Hassel. A Framework for Object-Oriented On-Line Analytic Processing. In *DOLAP '98 Proceedings of the 1st ACM international workshop on Data warehousing and OLAP*, pages 10–15. ACM, 1998.
- [4] Carlos Pampulim Caldeira. *Data Warehousing - Conceitos e Modelos*. Edições Sílabo, 2nd edition, 2012.
- [5] Wayne Eckerson. Gauge your data warehouse maturity. *DM Review*, 14(11):34, 2004.
- [6] Wayne Eckerson. Four Ways to Build a Data Warehouse. *The Data Administration Newsletter*, May 2007. <http://www.tdan.com/view-articles/4770>. [Acesso em: 26 setembro 2013].
- [7] The Eclipse Foundation. Eclipse IDE. <http://www.eclipse.org/>. [Acesso em: 16 setembro 2013].
- [8] The Kimball Group. Kimball core concepts. <http://www.kimballgroup.com/data-warehouse-business-intelligence-resources/kimball-core-concepts/>. [Acesso em: 18 Junho 2013].
- [9] The Kimball Group. Three critical components for successful self-service bi. <http://www.kimballgroup.com/2013/03/04/design-tip-153-three-critical-components-for-successful-self-service-bi/>, 2013. [Acesso em: 13 Agosto 2013].
- [10] Gabriele Höfling, Barbara Dinter, Carsten Sapia, and Markus Blaschka. The Olap Market: State of the Art and Research Issues. In *DOLAP '98 Proceedings of the 1st ACM international workshop on Data warehousing and OLAP*, pages 22–27. ACM, 1998.
- [11] IBM. Db2 Udb for iSeries. <http://publib.boulder.ibm.com/infocenter/iseriess/v5r3/index.jsp?topic=%2Frzatzc%2Frzatzcoverdb2.htm>. [Acesso em: 25 setembro 2013].

- [12] Gartner Inc. IT Glossary: Business Intelligence (BI). <http://www.gartner.com/it-glossary/business-intelligence-bi/>. [Acesso em: 7 Agosto 2013].
- [13] Gartner Inc. Magic quadrant for business intelligence and analytics platforms. <http://www.gartner.com/technology/reprints.do?id=1-1DZLPEP&ct=130207&st=sb>, 2013. [Acesso em: 17 Julho 2013].
- [14] Javlin Inc. CloverETL. <http://www.cloveretl.com/>. [Acesso em: 16 setembro 2013].
- [15] MicroStrategy Inc. The 5 styles of business intelligence. <http://www.microstrategy.co.kr/BI-Applications/5Styles/>. [Acesso em: 11 Abril 2013].
- [16] Pentaho Inc. Pentaho Data Integration (Kettle). <http://kettle.pentaho.com/>. [Acesso em: 16 setembro 2013].
- [17] Talend Inc. Talend Open Studio for Data Integration. <http://www.talend.com/>. [Acesso em: 16 setembro 2013].
- [18] William Harvey Inmon. *Building the Data Warehouse*. Wiley, 4th edition, 2005.
- [19] Ralph Kimball, Laura Reeves, Margy Ross, and Warren Thornthwaite. *The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses*. Wiley, Agosto 1998.
- [20] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit*. Wiley, 2nd edition, 2002.
- [21] MicroStrategy Inc. *Architecture for Enterprise Business Intelligence*.
- [22] MicroStrategy Inc. *Project Design Guide*, 9.3 edition.
- [23] MicroStrategy Inc. *System Administration Guide*, 9.3 edition.
- [24] Oracle Corporation (Sun Microsystems). Java Runtime Environment. <http://www.java.com/>. [Acesso em: 16 setembro 2013].
- [25] Margaret Rouse. Molap (multidimensional online analytical processing). <http://searchsqlserver.techtarget.com/definition/MOLAP>, 2005. [Acesso em: 20 Maio 2013].
- [26] Margaret Rouse. Multidimensional database (mdb). <http://searchoracle.techtarget.com/definition/multidimensional-database>, 2005. [Acesso em: 20 Maio 2013].
- [27] Efraim Turban, Ramesh Sharda, Dursun Delen, and David King. *Business Intelligence: A Managerial Approach*. Prentice Hall, 2nd edition, 2010.
- [28] Hugh J. Watson and Barbara H. Wixom. The Current State of Business Intelligence. *IEEE Computer Society*, 40:96–99, 2007.

- [29] Robert Wrembel and Christian Koncilia. *Data Warehouses and OLAP: Concepts, Architectures and Solutions*. IRM Press, 2007.